HIGHWAY INFORMATION SYSTEM
RELEASE 3.0
VOLUME 1
USER'S MANUAL

CE
EM

Department of Civil Engineering & Engineering Mechanics   Montana State University   Bozeman

HIGHWAY INFORMATION SYSTEM
RELEASE 3.0
VOLUME 1
USER'S MANUAL

Prepared for the

STATE OF MONTANA
DEPARTMENT OF HIGHWAYS
PLANNING AND RESEARCH BUREAU

In cooperation with the

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL HIGHWAY ADMINISTRATION

The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of the Montana Department of Highways or the Federal Highway Administration

Prepared by

Larry J. Coats, Glen L. Martin, Scott H. Danforth, and Michael J. Meldahl

DEPARTMENT OF CIVIL ENGINEERING AND ENGINEERING MECHANICS
MONTANA STATE UNIVERSITY
Bozeman, Montana 59715

March, 1974

TABLE OF CONTENTS

Table of Contents (Continued)

Table of Contents (Continued)

Chapter | Page

Table of Contents (Continued)

Table of Contents (Continued)

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This publication describes the use of release 3.0 of the Highway Information System (HIS). It is intended for all potential users of HIS, regardless of technical background.

Release 3.0 is a batch-processing and remote batch-processing system, and no on-line support is available. In designing HIS, however, an attempt has been made to allow future on-line support with a minimum of conversion effort when computer hardware and software becomes available.

The Highway Information System is composed of data files and software. Software is available both for maintaining the system files and for producing reports and summaries from the files.

HIS files are designed to eliminate data redundancy by storing each data item in one and only one file. This organization helps promote data integrity, forces file compatibility, and simplifies file maintenance. The files are organized around a common reference method, allowing cross-reference of two or more files in producing reports and summaries.

A great deal of effort has been spent orienting HIS toward the user rather than toward the computer programmer. The user prepares a "command" indicating his needs, and submits the command with a minimal amount of IBM Job Control statements. The system can be used by persons unfamiliar with computers or with the internal workings of HIS.

Release 3.0 is implemented under IBM's OS/VS1 operating system. User input falls into three categories: (1) OS/VS1 Job Control Language

(JCL) statements, (2) HIS commands, and (3) input data. OS/VS1 JCL is required to instruct OS/VS1 to execute the Highway Information System. Commands are used to define the execution requirements to HIS. Input data is required by a few of the HIS routines, such as those updating data files.

The following signs and conventions, consistent with those used by IBM, have been adopted throughout these manuals:

1) Uppercase letters and punctuation marks (except for brackets and braces) must be coded.
2) Lowercase letters and terms represent information that must be supplied by the user.
3) Information contained within brackets [ ] represents an option that may be included or omitted, depending upon the user's requirements.
4) Options contained with braces { } represents alternatives, one (and only one) of which must be chosen.

<u>The Reference Post System</u>

Highway Information System files are organized around the reference post system. This system is a method for uniquely identifying roadway locations (milepoints), and consists of a set of non-uniformly spaced physical reference posts located along roadways. The reference posts, in general, are a mile apart, buy may vary considerably from this distance. The first marker of a route is numbered zero, and succeeding markers are numbered sequentially.

In order to uniquely specify a milepoint on a given route, two items are specified: the number of a reference post, and the distance from that reference post to the roadway location. The distance specified is positive if travel from the reference post to the milepoint is toward higher-numbered reference posts, and negative if travel is toward lower-numbered reference posts.

As an example of the use of the reference post system, a point located 0.348 miles beyond reference post number 146 toward reference post 147 is specified as milepoint 146+0.348. The point may also be referenced in relation to marker 147. If, for example, markers 146 and 147 are 1.459 miles apart, the point may be specified as 147-1.111.

To determine the distance between two milepoints, it is necessary to establish the location of all of the reference posts on a route by means of a common point. A "true mileage" file locates each reference post with respect to the beginning milepoint (000+0.000) of the route on which it is located.

To complete the key for HIS files, the route system and route number are joined together with the milepoint. The route system is a 1-character code:

I -- Federal Aid Interstate,
P -- Federal Aid Primary,
S -- Federal Aid Secondary,
U -- Federal Aid Urban, and
L -- Local.

The route number is a 5-digit number. The complete key provides a unique identification for every roadway location stored in Highway Information System files.

HIS Files

HIS contains a number of individual files. The files are totally compatible, and several files are often accessed simultaneously to derive information for summaries and reports.

The Roadlog File

The roadlog file contains physical roadway information pertaining to all federal aid and local system routes. The file is organized around the common key of HIS -- the route system, route number, and milepoint. Each record contains the key of a roadlog "section break" (a location at which a discontinuity occurs, such as a change in the physical characteristics of the roadway, a county line, a city limit, a major junction, etc.) and defines the section of road between that break and the next. When two routes are coincident, a cross-reference technique is used to prevent duplication of data. All of the sections within the coincident stretch are fully coded for one of the routes. The second route contains only a "coincident" record that specifies the beginning and ending milepoints on the first route of the coincident roadway.

The True Mileage File

The true mileage file provides the location of each reference post with respect to the beginning of the route on which it is located.

The Traffic File

The traffic file contains "average annual daily traffic" (AADT) counts at each count station for three separate years. Each count station is located by its milepoint.

The Accident Files

For efficient storage and access, accident data is stored in two separate files: a detail file with information pertaining to overall data on each accident and a vehicle file containing data on persons and vehicles involved in each accident. These files are organized around an assigned accident number rather than milepoint to insure a unique key and because not all accidents occur on routes with reference posts. For those accidents whose location is specified with a milepoint, a directory file is maintained to allow access by milepoint.

The Sufficiency File

The sufficiency file contains data elements indicating quality and conditions of roadway sections. Records are keyed by milepoint.

The Bridge File

The bridge file contains a record for each bridge in the state. Data elements within the record describe the type and condition of the bridges. Bridges are stored and accessed by milepoint location.

The Railroad Crossing File

The railroad crossing file contains data elements describing all of the railroad crossings in the state. Crossings are keyed by milepoint.

Using HIS

HIS is a user-oriented system which can be used by persons having little or no technical background with computers. The user simply prepares a command card, and submits it along with a minimal amount of OS/VS1 Job Control Language (JCL).

OS/VS1 JCL Statements

Because the Highway Information System is executed by the IBM OS/VS1 operating system, OS/VS1 must receive instructions from the user. The first JCL card submitted with each run must be a "JOB" card. This first card indicates the start of a job to the operating

system and contains accounting information. JOB cards are acquired through the Data Processing Center, and are not shown in detail in this publication.

Following the JOB card, the user places an EXEC statement and any DD statements required for the run. The EXEC statement specifies the name of a "cataloged procedure" stored within the computer's memory. This procedure contains a set of standard JCL. The following procedures are available:

```
HIS
HISRLG   -- Roadlog subsystem
HISTRAF  -- Traffic subsystem or true mileage subsystem
HISACC   -- Accident subsystem
HISMEMO  -- Accident subsystem
HISACD   -- Accident subsystem
HISACCA  -- Accident subsystem
HISACCM  -- Accident subsystem
HISSUFF  -- Sufficiency subsystem
HISBRID  -- Bridge subsystem
HISRRX   -- Railroad crossing subsystem
```

The documentation below illustrates which procedure to use for each command.

A SYSIN DD statement is always required, and is used for entering HIS commands. This card is coded as "//SYSIN DD *," indicating that the commands follow immediately. After the last command is placed a "/*" card.

An example of a complete run is:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
     One or more HIS commands placed here
/*
```

Additional JCL may be required for some programs and when required is indicated with the description of these commands.

HIS Commands

HIS command cards are identified by a colon (:) in column 1. All commands must contain this identification. Immediately following the colon is coded the name of a HIS program to be executed. Execution options are selected by means of parameters coded on the command card following the program name. Each parameter consists of a keyword and an option, separated by an equal sign (e.g., FILE=ROADLOG). The first parameter is separated by a comma from the program name. Additional parameters, if any, are separated by commas. The last parameter is followed by at least one blank. An example of a valid command containing two paramters is:

```
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=PRIMARY
```

An example of a command containing no parameters is:

```
:LIST-CITY-TABLE
```

If an option contains a blank or a comma, it must be enclosed in quotes. An example of such a command is:

```
:RESTART-MEMOS,LOCATION='JONES,THOMAS L'
```

Continuation cards

When a command is too large to be contained on a single card, it may be continued on another by placing a comma after a complete parameter, leaving the remainder of the card blank. The continuation card must contain a colon in column 1, followed by one or more blanks and the continuation. An example of a valid command with continuation is:

```
:SURF-TYPE,REPORT=ROADLOG,DATA=SECONDARY,
:     SUMMARY=RTE-NO,MILEAGE=ALL,
:     TABLE-NUMBER=6
```

Comment cards

Comments may be included in a group of commands by including a card with a "greater-than" sign (>) in column 1. Comment cards are printed with the command listing, but are not processed.

The DATA, START-MILEPOINT and END-MILEPOINT Parameters

Of special note on HIS commands is the means of selecting roadways for processing. For many applications it is important to be able to select one or more entire route systems, one or more routes, or a portion of one route for processing.

A single route system may be selected by coding DATA=INTERSTATE, DATA=PRIMARY, DATA=SECONDARY, DATA=URBAN, or DATA=LOCAL. The interstate and primary systems may be combined by coding DATA=INTERSTATE+PRIMARY. The four federal aid systems may be combined by coding DATA=FEDERAL-AID. All five systems may be combined by coding DATA=ALL. To process a single route, the route system and number are specified, as in DATA=PRIMARY=5. A sequence of routes within a route system may be selected as in DATA=PRIMARY=3-6. To select only a portion of a route, the route is selected in the DATA parameter, and the starting and/or ending milepoints coded in the START-MILEPOINT and/or END-MILEPOINT parameters. For example, the combination DATA=PRIMARY=1,END-MILEPOINT=050+0.000 processes primary route 1 from its beginning up to and including milepoint 050+0.000. DATA=PRIMARY=1,START-MILEPOINT=100+0.000,END-MILEPOINT=200+0.000 processes route 1 from reference post 100 through reference post 200.

Formatting Printed Output

Users of the Highway Information System have considerable control over the format of printed output. Formatting options are implemented in two ways: (1) parameters coded on the EXEC card, and (2) parameters coded on HIS commands. In general, parameters coded on the EXEC card are in effect for the entire run, and parameters coded on a command are in effect only for that command.

Parameters can also be placed in effect for an entire run by coding them on a command that specifies "SYS-PARAM" as its program name.

Varying the Page Size

The user may specify the number of lines printed on each page by coding PAGESIZE=n on a command. The value coded may range from 1 to 999. Unless otherwise specified by coding PAGESIZE on a SYS-PARAM command, a pagesize of 60 is used. An example of PAGESIZE is:

```
// JOB
// EXEC HIS
//SYSIN DD *                    Notes:
:program-a                      Page size = 60
:program-b,PAGESIZE=46          Page size = 46
:program-c                      Page size = 60
:SYS-PARAM,PAGESIZE=30
:program-d                      Page size = 30
:program-e,PAGESIZE=40          Page size = 40
:program-f                      Page size = 30
/*
```

Printing Page Numbers

Page numbering may be started by coding PAGE-NUMBER=n on a command (n may range from 1 to 99999). Once begun, page numbering continues from one command to the next until a command specifying PAGE-NUMBER=STOP is executed. A third form of the PAGE-NUMBER parameter, PAGE-NUMBER=$+n, indicates that the page number is to be incremented by "n" before beginning

the next command. PAGE-INCREMENT=n (n ranging from 1 to 9) may be coded if page numbers are to be incremented by a value other than 1. Page numbers are always printed at the top of the page, and are printed in printer columns 120-124 unless otherwise indicated. The starting location of either even or odd page numbers (or both) may be set by coding EVEN-PAGE-POSITION=n or ODD-PAGE-POSITION=n (n may range from 1 to 128).

Printing Table Numbers

The user may specify a heading of the form "TABLE NUMBER n" printed at the top of each page of output generated by a command by coding TABLE-NUMBER=n (n ranging from 1 to 999) on that command. Once begun, table numbering continues to following commands, with the value n incremented by one for each command. To terminate table numbering, code TABLE-NUMBER=STOP.

Generating Top Margins

A top margin of "n" blank lines may be generated on each page of output by a command by specifying TOP-MARGIN=n on the command. If a top margin is set through SYS-PARAM, it may be returned to zero by coding TOP-MARGIN=NO.

Suppressing Page Ejection

Normally, HIS ejects to a new page when beginning a new command. The parameter PAGE-EJECT=SUPPRESS may be coded to suppress page ejection. If PAGE-EJECT=SUPPRESS is coded for the first command, that program's output begins on the same page as the command listing. If PAGE-EJECT=SUPPRESS is set in effect for the duration of a run on a SYS-PARAM command, it may be overridden by coding PAGE-EJECT=NOSUPPRESS.

Placing Output on Special Forms

The PRINTER-DD parameter is available for placing output on special forms. Two options may be used without requiring additional OS/VS1 JCL:

Coded Option	Paper to be printed on
PRINTER-DD=GREEN3	14x11 3-part green-striped paper.
PRINTER-DD=WHITE1	14x11 white paper.

To use forms other than these, OS/VS1 must be informed of an output class and form number (a list of those available may be obtained from the Data Processing Center). If all output in a given run is placed on the same form, code the parameter FORM='(class,,form)' on the EXEC card. Otherwise, prepare an OS/VS1 DD statement indicating the desired form, and code the name of this DD statement in the PRINTER-DD parameter. An example of the first method is:

```
// JOB
// EXEC HIS,FORM='(D,,0861)'
//SYSIN DD *
:LIST-CITY-TABLE
/*
```

An example of the second method is:

```
// JOB
// EXEC HIS
//OUTFORM DD SYSOUT=(D,,0861),
//   DCB=(BLKSIZE=1330,LRECL=133,RECRM=FBA)
//SYSIN DD *
:LIST-CITY-TABLE,PRINTER-DD=OUTFORM
/*
```

Generating Multiple Copies

The user may obtain more than one copy of an entire run by coding COPIES=n on the EXEC statement. When utilizing a PRINTER-DD parameter with a user-supplied DD statement, the parameter COPIES=n is coded on the DD statement rather than on the EXEC statement.

Storing Output on Tape or Disk

The OUTPUT-FILE parameter allows the user to save printed output for future printing. An OS DD statement defining a tape or disk output file is required. For full information on DD statements, see the IBM publication: OS/VS JCL Reference. An example of OUTPUT-FILE is:

```
// JOB
// EXEC HIS
//OUTFILE DD UNIT=TAPE,VOL=SER=026432,DISP=(,KEEP),
//   DSNAME=HIS.OUTPUT,
//   DCB=(BLKSIZE=1330,LRECL=133,RECFM=FB)
//SYSIN DD *
:LIST-CITY-TABLE,OUTPUT-FILE=OUTFILE
/*
```

The tape file is created in addition to the copy printed. After executing this run, additional copies of the output may be obtained through the use of procedure HISX133:

```
// JOB
// EXEC HISX133,PARM=3,FORM='(D,,0861)'
//SYSIN DD UNIT=TAPE,VOL=SER=026432,DISP=OLD,
//   DSNAME=HIS.OUTPUT
```

PARM=3 specifies three copies. FORM='(D,,0861)' specifies the output form. Either of these may be omitted, defaulting to PARM=1 and FORM=A.

Printing User Titles

The user may specify up to five 80-character lines to be printed on each page of output as a heading. The parameter TITLE-DD is used to enter the name of a DD statement defining the titles. An example of TITLE-DD usage is:

```
// JOB
// EXEC HIS
//SYSIN DD *
:LIST-CITY-TABLE,TITLE-DD=TITLES
/*
//TITLES DD *
     LISTING OF CITY TABLE, 4-1-73
/*
```

Data Privacy and Security

Privacy and security are important considerations in building an information system. Data privacy is necessary to prevent unauthorized access to sensitive data. Data security is necessary to prevent unauthorized modification to data. Both of these considerations are implemented within the Highway Information System by means of passwords protecting programs from unauthorized use. Those programs whose outputs contain sensitive data and those programs having the capability of updating or in any way modifying data files cannot be executed unless the appropriate password is included.

CHAPTER 2

ROADLOG SUBSYSTEM

Introduction

The roadlog file describes physical characteristics of roadways. Data is stored in the file according to milepoint, as described in Chapter 1. Each record contains information pertaining to a section of roadway. A "roadlog section" is identified by the beginning milepoint of the section. Section "breaks" occur at major junctions, city limits, county lines, and at any point at which the physical characteristics of the road (such as surface type or roadway width) change.

Roadlog Record Coding

There are two major categories of records in the roadlog file: mileage records and descriptor records. Mileage records contain the physical and administrative data for roadway sections. Descriptor records provide additional descriptions and information necessary for producing reports. The record type is identified by means of a "remark" code stored within the record. Five mileage types may be coded:

Blank	Mileage record
NE	Mileage record (non-existent section)
OS	Mileage record (out of state section)
SP	Mileage record (spur)
LP	Mileage record (loop)

Five types of descriptor records may be coded:

DS	Additional description for spurs, loops, and signed route numbers
ER	Other additional descriptions
EN	End of route
CO	Coincident
IL	Interstate loop

Detailed information on coding data cards is contained in Appendix A.

Tables

A number of programs utilize tables stored within a disk library. For example, a table is included that provides the names of Montana incorporated cities, and is accessed by programs that print city names in reports or summaries.

City Name Table

The city name table contains the city names, populations, and county numbers for the incorporated cities of Montana. The table contains one entry for each city, containing:

City name, left justified
City name, centered
City name, left justified with imbedded blanks replaced with hyphens
Population
County number (alphabetical numbering system)
County number (vehicle registration numbering system)

The population value is a one-digit code:

1	0-999
2	1,000-2,499
3	2,500-4,999
4	5,000-9,999
5	10,000-24,999
6	25,000-49,999
7	50,000 and over

A listing of the city table may be obtained with the LIST-CITY-TABLE program:

```
// JOB
// EXEC HIS
//SYSIN DD *
:LIST-CITY-TABLE
/*
```

Entries in the city table may be updated with program UPDATE-CITY-TABLE. Data cards are prepared in the following format:

1-3	City number (001-126).
4-21	Name, left justified.
22-39	Name, centered.
40-57	Name, left justified with hyphens.
58	Population.
59-60	County number, alphabetical.
61-62	County number, registration.
63-80	Blank.

The city number identifies the entry being updated, and is always coded. The remaining fields need to be coded only if they are being altered. An example of UPDATE-CITY-TABLE is:

```
// JOB
// EXEC HIS,DISP=OLD
//SYSIN DD *
:UPDATE-CITY-TABLE,DDNAME=UPDCITY
/*
//UPDCITY DD *
     data cards in order by city number
/*
```

<u>County Name Table</u>

The county name table contains county names, financial districts, Highway Patrol divisions, and county numbers for Montana counties. The table contains one entry for each county, containing:

County number (alphabetical numbering system)
County number (vehicle registration numbering system)
Financial district
Name, left justified
Name, centered
Name, left justified with imbedded blanks replaced
with hyphens
Highway Patrol division

To list the county table, use LIST-COUNTY-TABLE:

```
// JOB
// EXEC HIS
//SYSIN DD *
:LIST-COUNTY-TABLE
/*
```

To update entries in the county table, prepare data cards in the following format:

1-2	County number, alphabetical system.
3-4	County number, registration system.
5-6	Financial district.
7-21	Name, left justified.
22-36	Name, centered.
37-51	Name, left justified with hyphens.
52	Highway Patrol division.
53-80	Blank.

The alphabetical county number is always coded, and is used to identify the entry being updated. Remaining fields need to be coded only if they are being changed. An example of UPDATE-COUNTY-TABLE is:

```
// JOB
// EXEC HIS,DISP=OLD
//SYSIN DD *
:UPDATE-COUNTY-TABLE,DDNAME=UPDCNTY
/*
//UPDCNTY DD *
     data cards in order by alphabetical county number
/*
```

Surface Type Table

The surface type table contains a list of valid surface types in the roadlog file, and the corresponding surface type classification. A listing of the table may be obtained with:

```
// JOB
// EXEC HIS
//SYSIN DD *
:LIST-SURFACE-TABLE
/*
```

The surface type classification is a one-digit number:

1 -- Primitive
2 -- Unimproved
3 -- Graded and drained
4 -- Gravel
5 -- Bituminous surface treatment
6 -- Road mix
7 -- Plant mix
8 -- Portland cement concrete

Entries may be rewritten, inserted, or deleted by program UPDATE-SURFACE-TABLE. Data cards are prepared in the format:

1-4	Surface type.
5	Surface type classification.
6-80	Blank.

To delete an entry, code its surface type and leave the classification blank. To add an entry, code both the type and its classification. To rewrite an existing entry, code its surface type and its new classification. The data cards are submitted to UPDATE-SURFACE-TABLE:

```
// JOB
// EXEC HIS,DISP=OLD
//SYSIN DD *
:UPDATE-SURFACE-TABLE,DDNAME=UPDSURF
/*
//UPDSURF DD *
     data cards in any order
/*
```

Project Class Table

The project class table contains a list of projects and their corresponding administration codes, as used in the roadlog file. A listing of the table may be obtained with LIST-PROJECT-TABLE:

```
// JOB
// EXEC HIS
//SYSIN DD *
:LIST-PROJECT-TABLE
/*
```

Entries in the table may be revised, added, or deleted through program UPDATE-PROJECT-TABLE. Data cards are prepared in the following format:

1-2	Administration code.
3-6	Project class.
7-80	Blank.

To delete an entry, code the administration code and leave the project class blank. To add or rewrite an entry, code both the administration code and the corresponding project class. Submit the data cards to UPDATE-PROJECT-TABLE:

```
// JOB
// EXEC HIS,DISP=OLD
//SYSIN DD *
:UPDATE-PROJECT-TABLE,DDNAME=UPDPROJ
/*
//UPDPROJ DD *
     data cards in any order
/*
```

Roadlog File Maintenance

The following programs are implemented for maintaining the roadlog file:

DUMP	Prints selected records in data card format
LIST	Prints selected records in tabular format
LIST-ILOOPS	Prints summary of all Interstate loops
UPDATE	Deletes, inserts, and rewrites records
COPY	Generates backup copy of the file
CREATE	Reloads the file from a backup copy
REORGANIZE	Reformats the file by generating a backup copy

DUMP

DUMP produces a listing in data card format of those records specified. Required parameters are:

FILE=ROADLOG

$$\text{DATA=}\left\{\begin{array}{l}\left\{\begin{array}{l}\text{INTERSTATE}\\\text{PRIMARY}\\\text{SECONDARY}\\\text{URBAN}\\\text{LOCAL}\end{array}\quad\left[\text{=n}\ \left[\text{-n}\right]\right]\right\}\\\left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\\text{FEDERAL-AID}\\\text{ALL}\end{array}\right\}\end{array}\right\}$$

Optional parameters are:

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

Sample DUMP commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:DUMP,FILE=ROADLOG,DATA=INTERSTATE
:DUMP,FILE=ROADLOG,DATA=SECONDARY=209
:DUMP,FILE=ROADLOG,DATA=PRIMARY=3-6
:DUMP,FILE=ROADLOG,DATA=PRIMARY=1,
:    START-MILEPOINT=050+0.000,END-MILEPOINT=075+0.000
:DUMP,FILE=ROADLOG,DATA=PRIMARY=2,END-MILEPOINT=050+0.000
/*
```

LIST

LIST prints a formatted listing of the roadlog records selected.

Required parameters are:

FILE=ROADLOG

$$\text{DATA=}\left\{\begin{array}{l}\left\{\begin{array}{l}\text{INTERSTATE}\\\text{PRIMARY}\\\text{SECONDARY}\\\text{URBAN}\\\text{LOCAL}\end{array}\right\}\left[\text{=n}\ \left[\text{-n}\right]\right]\\\left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\\text{FEDERAL-AID}\\\text{ALL}\end{array}\right\}\end{array}\right\}$$

Optional parameters are:

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

Some sample commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=PRIMARY
:LIST,FILE=ROADLOG,DATA=INTERSTATE=90,
:    START-MILEPOINT=005+0.000,END-MILEPOINT=020+0.000
/*
```

LIST-ILOOPS

LIST-ILOOPS aids in coding Interstate loops. The primary portion of the roadlog file is scanned for IL type records (these records identify Interstate loop locations). Each time an IL record is read, the records specified are read and listed. In addition, the loop length, total urban mileage, total mileage outside of federal reservations, and total other mileage is printed. The only required parameter is FILE=ROADLOG. An example of a LIST-ILOOPS run is:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:LIST-ILOOPS,FILE=ROADLOG
/*
```

UPDATE

UPDATE is used to update the roadlog file. Four transaction types are available, and the user must keep the four data card types separate. A separate command is entered for each transaction type within a given run. Any number of updates within a transaction type may be submitted with one command. Parameters that must always be coded on an UPDATE command are:

FILE=ROADLOG

DDNAME=name

$$\text{FUNCTION=} \left\{ \begin{array}{l} \text{DELETE} \\ \text{INSERT} \\ \text{REWRITE} \\ \text{NEW-KEY} \end{array} \right\}$$

An additional parameter, DEBUG=1, may optionally be coded. This parameter is intended to aid the user in the event of a faulty edit test, or in the event that additions to roadlog records result in faulty edits. When DEBUG=1 is specified, records are edited as usual and any error messages are printed, but the transaction takes place even if severe errors are detected. DEBUG=1 has no meaning when coded with FUNCTION=DELETE or FUNCTION=NEW-KEY.

The DDNAME parameter is used to link the user's data cards to the UPDATE command. Any name of eight characters or less that does not occur within the cataloged procedure HISRLG may be coded. If unfamiliar with the contents of HISRLG, the following names can be safely used:

```
UPDRLGD -- DELETE
UPDRLGI -- INSERT
UPDRLGN -- NEW-KEY
UPDRLGR -- REWRITE
```

The same name coded in the DDNAME parameter is used on a "DD *" Job Control Statement, after which are placed the user's data cards. A "/*" card is placed after the last data card. Some examples of UPDATE runs are:

```
// JOB
// EXEC HISRLG,DISP=OLD
//SYSIN DD *
:UPDATE,FILE=ROADLOG,FUNCTION=INSERT,DDNAME=ADDRECS
:UPDATE,FILE=ROADLOG,FUNCTION=REWRITE,DDNAME=CHANGE
/*
//ADDRECS DD *
     Data cards for INSERT function
/*
//CHANGE DD *
     Data cards for REWRITE function
/*
```

When updating, each record is edited for possible data errors. Two types of errors may be detected: "severe errors" that cause rejection of data cards, and "warnings" that indicate a possible error but do not cause data card rejection. Severe error messages are printed in the format "***** (E) message *****." Warnings are printed in the format "***** (W) message *****." A complete list of error messages is included in Appendix A.

The DELETE Function

The DELETE function is used to delete records from the file. Data cards are prepared in the format:

1-15	Key of record being deleted
16-80	Blank

The INSERT Function

The INSERT function is used to add records to the file. Data cards are prepared as directed in Appendix A. When inserting descriptor records, only an "A" card is used. When inserting mileage records, an "A" and a "B" card are always used, and a "C" card is optional.

The REWRITE Function

The REWRITE function is used to alter any fields within records other than the key field. Data cards are prepared as directed in Appendix A, coding the key fields and those fields being altered. When rewriting descriptor records, only "A" cards are used. When inserting mileage records, any combination of "A," "B," and "C" cards are used. When a character field is to be filled with blanks, code the field with dollar signs. When a numeric field is to be set to zero, the field may be coded with zeroes or dollar signs.

The NEW-KEY Function

The NEW-KEY function is used to alter a record's key field. Data cards are prepared in the format:

1-15	Key of existing record
16	Equal sign
17-31	New key
32-80	Blank

COPY

COPY is used to generate a backup copy of the roadlog file. Use of COPY requires a knowledge of OS/VS1 Job Control Language, as a DD statement must be provided to allocate the backup file. The DCB parameters LRECL=160 and RECFM=FB are always coded. The block-size may be chosen by the user. An example of COPY is:

```
// JOB
// EXEC HISRLG,DISP=OLD
//SAVERLG DD UNIT=(TAPE,,DEFER),VOL=SER=012345,
//           DISP=(NEW,KEEP),DSNAME=HIS.ROADLOG.BACKUP,
//           DCB=(BLKSIZE=16000,LRECL=160,RECFM=FB)
//SYSIN DD *
:COPY,FILE=ROADLOG
/*
```

CREATE

CREATE is used to reload the file from a previous backup copy. A SAVERLG DD statement is supplied indicating the backup file location. An example run is:

```
// JOB
// EXEC HISRLG,DISP=OLD
//SAVERLG DD UNIT=(TAPE,,DEFER),VOL=SER=012345,
//             DISP=OLD,DSNAME=HIS.ROADLOG.BACKUP
//SYSIN DD *
:CREATE,FILE=ROADLOG
/*
```

REORGANIZE

Deletion, insertion, and key revision updates result in wasted storage space, and as a result the file must be periodically reorganized. Reorganization is performed by generating a backup copy of the file and reloading, and can also be done with COPY and CREATE. A sample run is:

```
// JOB
// EXEC HISRLG,DISP=OLD
//SAVERLG DD UNIT=(TAPE,,DEFER),VOL=SER=123456,
//             DISP=(NEW,KEEP),DSNAME=HIS.ROADLOG.BACKUP,
//             DCB=(BLKSIZE=16000,LRECL=160,RECFM=FB)
//SYSIN DD *
:REORGANIZE,FILE=ROADLOG
/*
```

The Annual Federal Aid Roadlog Report

The following programs are used in producing the annual Federal Aid Roadlog report:

LIST-&-SUM	Prints the main listing of roadlog data
SURF-TYPE	Prints a set of summaries according to surface type

SUMMARY-BY-ROUTES	Prints summaries according to route number
SUMMARY-BY-LOCATION	Prints summaries according to route number
FORHWY-SUMMARY	Prints summaries of forest highway routes

In addition to their use in producing the annual report, these programs are helpful in roadlog file maintenance and in data access. Some programs provide certain data editing functions that cannot be reasonably performed by UPDATE, and prints error messages as required.

LIST-&-SUM

LIST-&-SUM prints the main listing of roadlog data that forms the bulk of the annual Federal Aid Roadlog report. Two sets of output are produced for each route processed: a formatted listing of selected data fields from the roadlog records, and a summary of mileage by county. Required parameters on LIST-&-SUM commands are:

REPORT=ROADLOG

$$\text{DATA=}\left\{\begin{array}{l}\left\{\left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL}\end{array}\right. \quad \left[=\text{n} \quad [-\text{n}]\right]\right\}\\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL}\end{array}\right\}\end{array}\right\}$$

Optional parameters available for LIST-&-SUM are:

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

Examples of LIST-&-SUM commands together with required OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIMARY
:LIST-&-SUM,REPORT=DATA=SECONDARY=201-300
:LIST-&-SUM,REPORT=ROADLOG,DATA=INTERSTATE=90,
:   START-MILEPOINT=200+0.000,END-MILEPOINT=300+0.000
/*
```

LIST-&-SUM provides limited data editing functions. The following error messages may be printed:

***** INVALID COUNTY IN ABOVE RECORD *****

> A county number appears in a mileage record having a value other than 1 through 56.

***** NO RECORD FOR COINCIDENT SECTION *****

> A roadlog CO record specified a beginning key that does not exist in the file.

***** CONSTRUCTED MILEAGE IN ERROR: total *****

> The total constructed mileage for a route, as printed in this message, does not equal the value printed in the county summary. The value in the county summary is the total section length, less wye and unimproved mileage.

***** INSUFFICIENT STORAGE FOR COUNTY SUMMARY *****

> A route has mileage in more counties than is provided for in the program. Should this message appear, LIST-&-SUM must be modified. Instructions for increasing the storage allocation are found in <u>Highway Information System Release 3.0: Programming Details</u>

<u>SURF-TYPE</u>

SURF-TYPE is capable of producing a large number of different summaries according to surface type and a parameter chosen by the user. Required parameters on SURF-TYPE commands are:

```
REPORT=ROADLOG

           ⎧ RTE-NO ⎫
           ⎪ YR-BLT ⎪
           ⎪ YR-IMP ⎪
SUMMARY=   ⎨ SUR-WD ⎬
           ⎪ COUNTY ⎪
           ⎪ CITIES ⎪
           ⎩ PROJ-# ⎭

        ⎧ ⎧ INTERSTATE                    ⎫ ⎫
        ⎪ ⎪ PRIMARY                       ⎪ ⎪
        ⎪ ⎨ SECONDARY    [=n  [-n]]       ⎬ ⎪
        ⎪ ⎪ URBAN                         ⎪ ⎪
DATA=   ⎨ ⎩ LOCAL                         ⎭ ⎬
        ⎪                                   ⎪
        ⎪ ⎧ INTERSTATE+PRIMARY ⎫            ⎪
        ⎪ ⎨ FEDERAL-AID        ⎬            ⎪
        ⎪ ⎪ ALL                ⎪            ⎪
        ⎩ ⎩ ILOOP              ⎭            ⎭
```

Optional parameters are:

```
MILEAGE=        URBAN
                ALL

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

Examples of SURF-TYPE commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:SURF-TYPE,REPORT=ROADLOG,SUMMARY=YR-BLT,
:    DATA=INTERSTATE+PRIMARY
:SURF-TYPE,REPORT=ROADLOG,SUMMARY=COUNTY,MILEAGE=URBAN,
:    DATA=PRIMARY=2,START-MILEPOINT=082+0.356,
:    END-MILEPOINT=142+0.778
/*
```

The SUMMARY parameter indicates the type of summary being printed. For example, SUMMARY=COUNTY prints a summary according to surface type and county. The various options are:

RTE-NO -- Route number
YR-BLT -- Year built
YR-IMP -- Year improved
SUR-WD -- Surface width
COUNTY -- County
CITIES -- Cities
PROJ-# -- Project classification

The MILEAGE parameter may be used to limit processing to urban records. MILEAGE=ALL is assumed in the absence of a MILEAGE parameter.

The DATA=ILOOP option is provided to allow summaries of Interstate loop mileage.

SURF-TYPE performs several data editing functions during processing. Messages that may be printed are:

***** UNKNOWN SURFACE TYPE. KEY=key,SURF-TYPE=surface-type *****

A mileage record contains an invalid surface type. All surface types must be stored in the surface type table documented above. If the surface type is valid, it must be added to the surface type table. Otherwise, the roadlog record must be updated with a correct surface type.

***** PARAMETER HAS ZERO VALUE AT KEY = key *****

The parameter (surface width, year built, etc.) has a zero value in the record whose key is printed.

***** INVALID ILOOP DESCRIPTION. KEY=key,DESCRIPTION=description *****

This message can be printed only when DATA=ILOOP is specified, and indicates that an invalid IL record has been read.

***** STORAGE ALLOCATION EXCEEDED *****

This message indicates a SURF-TYPE program error. Instructions on increasing storage allocation in the event of this error may be found in Highway Information System Release 3.0: Programming Details.

SUMMARY-BY-ROUTES

SUMMARY-BY-ROUTES provides a summary broken down by route number and type of mileage. One line is printed for each route in the route system(s) specified, showing the following mileage types:

Route mileage
Constructed mileage
Unimproved mileage
Wye mileage
Municipal mileage
County mileage

National forest mileage
Indian reservation mileage
Game reserve mileage
State forest mileage
National park mileage
State park mileage
National monument mileage
Military reservation mileage

The following parameters are mandatory on SUMMARY-BY-ROUTES commands:

REPORT=ROADLOG

DATA= { INTERSTATE | PRIMARY | SECONDARY | URBAN | INTERSTATE+PRIMARY | FEDERAL-AID }

Sample commands with OS/VS1 Job Control are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=INTERSTATE+PRIMARY
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=SECONDARY
/*
```

SUMMARY-BY-LOCATION

SUMMARY-BY-LOCATION provides three summaries broken down by route number and by location (inside or outside of federal reservations). When DATA=INTERSTATE or DATA=SECONDARY is specified, the summaries show route lengths. Otherwise, the summary shows section lengths. DATA=INTERSTATE+PRIMARY also results in printing the status of the 7% system, which includes all mileage on the Inter-

state and Primary systems located outside federal reservations, excepting Interstate loops and urban mileage. Required parameters on SUMMARY-BY-LOCATION commands are:

```
REPORT=ROADLOG

        ( INTERSTATE         )
DATA=   { INTERSTATE+PRIMARY }
        ( SECONDARY          )
```

OS/VS1 Job Control cards and SUMMARY-BY-LOCATION sample commands are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:SUMMARY-BY-LOCATION,DATA=INTERSTATE
:SUMMARY-BY-LOCATION,DATA=INTERSTATE+PRIMARY
/*
```

<u>FORHWY-SUMMARY</u>

FORHWY-SUMMARY provides two summaries, the first a breakdown of forest highway mileage by location, and the second a breakdown of forest highway mileage by surface type. Required parameters are:

```
REPORT=ROADLOG

             ( LOCATION  )
FHSUMMARY=   { SURF-TYPE }
```

A sample set of commands with OS/VS1 JCL is:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=SURF-TYPE
:FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=LOCATION
/*
```

The Annual State Mileage Tables

Two programs are implemented for producing the annual State Mileage Tables. These are:

STATE-MILEAGE-502 -- 502 report form
STATE-MILEAGE-505 -- 505 report form

STATE-MILEAGE-502

This program provides a breakdown of mileage according to surface type and rural-municipal-urban location. The surface types are shown under seven categories (which differ from those used in the annual Federal Aid Roadlog report):

Category	Surface Types
A	0001
B	0002
C	0010,0011
E	2010
F,G-1,H-1	3210,4221,4241
G-2,H-2,I,K,L,M	4131,4134,4154,4231,4251,4252,4253,4254 4706,4716,6101,6201,6202,6203,6211,6706 6805,6806,6807,8301,9462,9662,9676,9740 9742
J	7001,7104,7201,7202,7204,7211,7212,7221 7222,7708

Roadlog records having a municipal location code are shown under the heading "municipal." All others are shown under "rural." In addition, all records specifying a city of population 5,000 or greater are shown under "urban."

The only parameter required on STATE-MILEAGE-502 commands is:

$$\text{DATA=}\left\{\begin{array}{l}\left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL}\end{array}\quad\left[\text{=n}\ \left[\text{-n}\right]\right]\right\}\\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL}\end{array}\right\}\end{array}\right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Sample commands with OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:STATE-MILEAGE-502,DATA=PRIMARY=8-12
:STATE-MILEAGE-502,DATA=LOCAL
/*
```

STATE-MILEAGE-505

STATE-MILEAGE-505 prints a breakdown of mileage according to surface type, surface width, average daily traffic, and rural-municipal-urban location. Only surfaced roadways are shown in the summary; categories A, B and C are not included. The HIS traffic file is accessed for traffic information.

Parameters on STATE-MILEAGE-505 commands are the same as for STATE-MILEAGE-502. A sample run is:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:STATE-MILEAGE-505,DATA=SECONDARY
:STATE-MILEAGE-505,DATA=PRIMARY=1,START-MILEPOINT=200+0.000
/*
```

CHAPTER 3

TRUE MILEAGE SUBSYSTEM

Introduction

The true mileage file pinpoints the locations of reference posts on the federal aid and local system roadways. One record is stored for each reference post, and specifies the distance from it to the beginning of the route on which it is located.

True Mileage Record Coding

The true mileage file utilizes a 9-character field for its key, rather than the 15-character field utilized by other HIS files. The key consists only of the route system designation, route number, and reference post. Each record gives the true mileage for the reference post specified in its key. Detailed information on coding data cards is provided in Appendix B.

True Mileage File Maintenance

The following programs are implemented for maintaining the true mileage file:

LIST	Prints selected records.
UPDATE	Deletes, inserts, and rewrites records.

COPY	Generates backup copy of the file.
CREATE	Reloads the file from a backup copy.
REORGANIZE	Reformats the file by generating a backup copy and reloading.
ROADLOG-TRUMILE-EDIT	Cross-checks true mileage data with the roadlog file to "spot" invalid records.

LIST

LIST prints a formatted listing of the true mileage records specified. Required parameters are:

FILE=TRUMILE

$$\text{DATA=} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{INTERSTATE} \\ \text{PRIMARY} \\ \text{SECONDARY} \\ \text{URBAN} \\ \text{LOCAL} \end{array} \left[=\text{n} \left[-\text{n} \right] \right] \right\} \\ \left\{ \begin{array}{l} \text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID} \\ \text{ALL} \end{array} \right\} \end{array} \right\}$$

Optional parameters are:

START-MILEPOINT=xxx

END-MILEPOINT=xxx

Some sample commands with OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:LIST,FILE=TRUMILE,DATA=PRIMARY
:LIST,FILE=TRUMILE,DATA=LOCAL=14001-14999
:LIST,FILE=TRUMILE,DATA=PRIMARY=1,
:      START-MILEPOINT=050,END-MILEPOINT=100
/*
```

UPDATE

UPDATE is used to update the true mileage file. Four transaction types are available, and the user must keep the four data card types separate. A separate command is required for each transaction type within a given run. Any number of updates within a single transaction type may be submitted with one command. Parameters that are always coded on a true mileage UPDATE command are:

FILE=TRUMILE

DDNAME=name

$$\text{FUNCTION=} \left\{ \begin{array}{l} \text{DELETE} \\ \text{INSERT} \\ \text{REWRITE} \\ \text{SEQL-REWRITE} \end{array} \right\}$$

The DDNAME parameter is used to link the user's data cards to the UPDATE command. Any name of eight characters or less that does not occur within the cataloged procedure HISTRAF may be coded. If unfamiliar with HISTRAF, the following names may be safely used:

```
UPDTRMD -- DELETE
UPDTRMI -- INSERT
UPDTRMR -- REWRITE
UPDTRMS -- SEQL-REWRITE
```

The same name coded in the DDNAME parameter is used on a "DD *" Job Control statement, after which are placed user's data cards. A "/*" card is placed after the last data card. Some examples of UPDATE runs are:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SYSIN DD *
:UPDATE,FILE=TRUMILE,DDNAME=SUBTRACT,FUNCTION=DELETE
:UPDATE,FILE=TRUMILE,DDNAME=STICKIN,FUNCTION=INSERT
/*
//SUBTRACT DD *
     Data cards for DELETE function
/*
//STICKIN DD *
     Data cards for INSERT function
/*
```

When updating, records are edited for possible data errors. Two types of errors may be detected: "severe errors" that cause rejection of data cards, and "warnings" that indicate a possible error but do not cause rejection of data cards. Severe error messages are printed in the format "***** (E) message *****." Warnings are printed in the format "***** (W) message *****." A complete list of error messages is included in Appendix B.

The DELETE Function

The DELETE function is used to delete records from the file. Data cards are prepared in the format:

1-9	Key of record being deleted
10-80	Blank

The INSERT Function

The INSERT function is used to add new records to the file. Data cards are prepared in accordance with Appendix B.

The REWRITE Function

The REWRITE function is used to alter any fields within records other than the key field. Data cards are prepared as directed in Appendix B, coding the key field and those fields being altered.

The SEQL-REWRITE Function

The SEQL-REWRITE is available for adjusting a set of contiguous reference posts on a single route by a specified constant. Data cards are prepared in the following format:

1-7	Adjustment factor in form ±nn.nnn
8	Blank
9	Route system (I, P, S, U, or L)
10-14	Route number
15	Blank
16-18	Beginning reference post
19	Blank
20-22	Ending reference post
23	Blank
24-29	Effective date (two digits each for month, day, and year)
30-80	Blank

The adjustment factor is added to all records between the specified reference posts, inclusive. Optionally, the ending reference post field may be left blank and the adjustment factor is added to all records from the beginning reference post to the end of the route. The effective date is stored in each record updated.

COPY

COPY is used to generate a backup copy of the true mileage file. Use of COPY requires a knowledge of OS/VS1 Job Control Language. A DD statement must be provided to allocate the backup file on tape or disk. The DCB parameters LRECL=26 and RECFM=FB are always coded. The blocksize is chosen by the user. An example of COPY is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SAVETRM DD UNIT=(TAPE,,DEFER),VOL=SER=999999,
//            DISP=(NEW,KEEP),DSNAME=HIS.TRUMILE.BACKUP,
//            DCB=(BLKSIZE=7800,LRECL=26,RECFM=FB)
//SYSIN DD *
:COPY,FILE=TRUMILE
/*
```

CREATE

CREATE is used to reload the file from a previous backup copy. A SAVETRM DD statement is supplied to indicate the backup file location on tape or disk. A sample run is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SAVETRM DD UNIT=(TAPE,,DEFER),VOL=SER=999999,DISP=OLD,
//           DSNAME=HIS.TRUMILE.BACKUP
//SYSIN DD *
:CREATE,FILE=TRUMILE
/*
```

REORGANIZE

Deletion and insertion of records causes wasted storage space in the file. As a result, the file should be periodically reorganized. Reorganization is done by generating a backup copy of the file and reloading. If a current backup copy exists, reorganization can be done by using CREATE. Otherwise, REORGANIZE can be used, and a current backup copy of the file can be saved. A sample run is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SAVETRM DD UNIT=(TAPE,,DEFER),VOL=SER=999999,
//           DISP=(NEW,KEEP),DSNAME=HIS.TRUMILE.BACKUP,
//           DCB=(BLKSIZE=7800,LRECL=26,RECFM=FB)
//SYSIN DD *
:REORGANIZE,FILE=TRUMILE
/*
```

This program performs a cross-check between true mileage entries and roadlog route lengths. One line of output is printed for each roadlog record, including:

Roadlog key
Roadlog remark
Roadlog description
Accumulated roadlog route mileage
True mileage calculated from roadlog key using true mileage file
Difference between true mileage and accumulated route mileage, if any
Calculated roadlog milepoint, if roadlog and true mileage records disagree

The only parameter that is required on ROADLOG-TRUMILE-EDIT commands is:

```
        ┌ ┌ INTERSTATE                      ┐ ┐
        │ │ PRIMARY                         │ │
        │ { SECONDARY    [=n [-n]]          } │
        │ │ URBAN                           │ │
DATA=   { └ LOCAL                           ┘ }
        │ ┌ INTERSTATE+PRIMARY ┐              │
        │ { FEDERAL-AID        }              │
        └ └ ALL                ┘              ┘
```

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Examples of ROADLOG-TRUMILE-EDIT commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:ROADLOG-TRUMILE-EDIT,DATA=INTERSTATE
:ROADLOG-TRUMILE-EDIT,DATA=PRIMARY=1,END-MILEPOINT=025+0.999
/*
```

CHAPTER 4

TRAFFIC SUBSYSTEM

Introduction

The traffic file contains traffic data for the three years prior to the current year. The average annual daily traffic (AADT) for each year is stored with the key being the milepoint at which a count was taken or estimated. The percentage of the AADT which is out-of-state vehicles, pickups, and commercial vehicles is also stored in the record. In addition to the complete three years, a fourth field is included for filling in data during the current year.

The traffic file "key," like that of most other Highway Information System files, consists of the route system designation, route number, reference post and distance from reference post.

Traffic Record Coding

Two major categories of records are stored in the traffic file: traffic count records and descriptor records. The record type is indicated by a 1-character "remark" code within the record.

"Traffic count" records are those records containing average annual daily traffic counts at specific milepoints. The values coded may be an actual or an estimated value.

Traffic count records are further subdivided into two subgroups: major section break records and minor section break records. Major

section breaks occur at county lines, city limits, major junctions, and other locations at which breaks are desired in traffic reports. Minor section breaks are locations within a traffic section at which traffic counts have been measured or estimated.

When printing the annual Federal Aid Traffic by Sections report, descriptions and county numbers are retrieved from the roadlog file at each major section break. Hence, a roadlog mileage record with an identical key must exist for each major section break record in the traffic file. Minor section break records need not correspond to roadlog records.

The types of traffic count records and their corresponding remark codes are:

W -- Rural major break
T -- Municipal major break
O -- Out-of-state major break
N -- Non-existent major break
R -- Rural minor break
M -- Municipal minor break

Traffic descriptor records are used to identify the beginnings of spurs, loops, and coincident sections. The types of descriptor records are:

C -- Coincident descriptor
L -- Loop descriptor
S -- Spur descriptor

No traffic data is coded in descriptor records. A corresponding roadlog descriptor record with an identical key must exist for each traffic

descriptor record. A roadlog CO record must correspond to each traffic C record. A roadlog DS record, naming the spur or loop, must correspond to each traffic S and L record.

Detailed information on coding traffic records is provided in Appendix C.

Traffic Sections

Because each traffic count record contains data at a specific milepoint, at least two records are required to define a section. The first record of a section is a major break record (W, T, O, or N) defining the type of section. The last record of the section is another major section break record, which contains a code indicating the type of the next section. If no contiguous section follows (i.e., the end of the route is reached, or a coincident section or spur or loop follows), this last record contains the same code as the first record of the section. If the beginning record contains a W, T, or O code, any number of subsection records may be included within the section.

When a roadway discontinuity occurs, the following conventions are followed:

1. End of route.

   The last section of the route contains the same major section break code at beginning and end. No end-of-route indication is necessary.

2. Coincident section.

When a route is coincident with another route, traffic counts are coded only on one of the routes. A C record is coded on the second route at the point where coincidence begins, and must correspond to a CO record in the roadlog file. A major section break record, terminating the previous section, is coded .01 mile or less upstream from the C record.

3. Spur or loop.

Although reference posted contiguously with the main portion of a route, each spur or loop is a separate piece of roadway. The previous section is terminated with the proper section break code. An L or S descriptor record is placed between this termination record and the first traffic count record of the spur or loop.

Some examples of highway sections are:

Rural section

W -- Begins rural section
R -- Minor break
R -- Minor break
W -- Ends rural section

Municipal section

T -- Begins municipal section
M -- Minor break
T -- Ends municipal section

Non-existent section

N -- Begins non-existent section
N -- Ends non-existent section
(minor breaks are not allowed)

Two rural sections preceding a coincident section, and a coincident section followed by another rural section

W -- Begins first rural section
R -- Minor break
W -- Ends first rural section and begins second rural section
R -- Minor break
R -- Minor break
W -- Ends second rural section
C -- Begins coincident section
W -- Ends coincident section and begins third rural section
W -- Ends third rural section

Municipal section following rural section

W -- Begins rural section
R -- Minor break
T -- Ends rural section and begins municipal section
M -- Minor break
M -- Minor break
T -- Ends municipal section

Non-existent section preceded and followed by rural section

W -- Begins rural section
R -- Minor break
N -- Ends rural section and begins non-existent section
W -- Ends non-existent section and begins rural section
W -- Ends rural section

Rural loop

W -- Begins last section of route prior to loop
R -- Minor break
R -- Minor break
W -- Ends rural section
L -- Indicates start of loop
W -- Begins rural section
R -- Minor break
W -- Ends rural section

Coding Rural and Out-of-State Sections

The traffic file has space for four data years. Three positions are utilized for data pertaining to the three years preceding the current year. The fourth is used for updating the file during the year as counts are taken. Once a year these values are shifted. Normally, rural sections contain data counts for all three of the retained years. Counting substations are rarely abandoned, and when this occurs the corresponding record is deleted from the file. In the case of a newly-constructed road, however, three years pass before all three fields contain data. When a new road is built, the records in the file corresponding to the old section are deleted. New records with data for the new road for the latest year only are coded. When producing the Traffic by Sections report, the ADT's for the earlier years are computed only on the basis of coded data. If no data at all is coded within a section, no values are computed. If data is given at the beginning and end of a section (i.e., the new road comprises only a portion of the section), the values are computed and weighted as if the uncoded records were not present. (Note that these records are used for computing the values for the most recent year.) Records within rural and out-of-state sections always contain values for the most recent year and contain values for the other years if the road has been in existence long enough.

Municipal Sections

Traffic counts in municipal sections, unlike those for rural sections, need not be taken every year. Hence, only years in which data is actually taken need to be coded. Uncoded data anywhere in a section for a given year causes the Traffic by Sections program to skip over that section in calculating AADT's for that year. When a T record terminates a rural or out-of-state section, it must be coded in the same manner as a W record.

Non-existent Sections

No traffic counts are taken within non-existent sections. Hence, N records need not contain traffic counts unless they are used to terminate other sections.

Traffic File Maintenance

The following programs are implemented for maintaining the traffic file:

DUMP	Prints selected records in "dump" format
LIST	Prints formatted listing of selected records
KEY-LIST	Prints coding form for new year data
UPDATE	Deletes, inserts, and rewrites records

UPDATE-BY-YEAR	Shifts newest three years into oldest three year positions
COPY	Generates a backup copy of the file
CREATE	Reloads the file from a backup copy
REORGANIZE	Reformats the file by generating a backup copy and reloading

DUMP

DUMP prints the specified traffic records in a format similar to data cards, using one line per record. Required parameters are:

FILE=TRAFFIC

$$\text{DATA=}\left\{\begin{array}{l} \left\{\begin{array}{ll}\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL}\end{array} & \left[\text{=n}\ \left[\text{-n}\right]\right]\end{array}\right\} \\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL}\end{array}\right\}\end{array}\right\}$$

Optional parameters that may be coded are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Examples of DUMP commands and accompanying OS/VS1 Job Control statements are:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:DUMP,FILE=TRAFFIC,DATA=PRIMARY
:DUMP,FILE=TRAFFIC,DATA=INTERSTATE=90,END-MILEPOINT=100+0.000
/*
```

LIST

LIST provides a formatted listing of specified records. Data is printed for three years only. When LIST=1 is specified, the oldest three years are printed. When LIST=2 is specified, the newest three years are printed. Also printed from traffic records are the key, remark, future factor, and design hour volume. The true mileage is printed for each traffic count record. The corresponding roadlog description is printed for each major break and descriptor record. Required parameters on LIST commands are:

FILE=TRAFFIC

$$\text{LIST=}\quad \begin{array}{l} 1 \\ 2 \end{array}$$

$$\text{DATA=} \left\{ \begin{array}{l} \left\{ \left\{ \begin{array}{l} \text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL} \end{array} \right. \quad \left[\text{=n} \left[\text{-n} \right] \right] \right\} \\ \left\{ \begin{array}{l} \text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL} \end{array} \right\} \end{array} \right\}$$

Optional parameters are:

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

Examples of LIST commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:LIST,FILE=TRAFFIC,LIST=1,DATA=INTERSTATE
:LIST,FILE=TRAFFIC,LIST=2,DATA=PRIMARY=1-3
/*
```

KEY-LIST

KEY-LIST prints record keys onto a coding form on which may be coded updates for entering data for a new year. Required parameters on KEY-LIST commands are:

```
FILE=TRAFFIC
      {  { INTERSTATE                          }  }
      {  { PRIMARY                             }  }
      {  { SECONDARY     [ =n  [ -n ] ]        }  }
      {  { URBAN                               }  }
DATA= {  { LOCAL                               }  }
      {  { INTERSTATE+PRIMARY }                   }
      {  { FEDERAL-AID        }                   }
      {  { ALL                }                   }
```

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

A sample job setup for KEY-LIST is:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:KEY-LIST,FILE=TRAFFIC,DATA=INTERSTATE+PRIMARY
:KEY-LIST,FILE=TRAFFIC,DATA=URBAN
/*
```

UPDATE

UPDATE is used to update the file. Four transaction types are available, and the user must keep the four data card types separate. A separate command is entered for each transaction type within a given run. Any number of updates within a transaction type may be submitted with one command. Parameters that must always be coded on an UPDATE command are:

FILE=TRAFFIC

DDNAME=name

$$\text{FUNCTION=}\left\{\begin{array}{l}\text{DELETE}\\\text{INSERT}\\\text{REWRITE}\\\text{NEW-KEY}\end{array}\right\}$$

The DDNAME parameter is used to link the user's data cards to the UPDATE command. Any name of eight characters or less that does not occur within the cataloged procedure HISTRAF may be coded. If unfamiliar with the contents of HISTRAF, the following names can be safely used:

```
UPDTRFD -- DELETE
UPDTRFI -- INSERT
UPDTRFN -- NEW-KEY
UPDTRFR -- REWRITE
```

The same name is used to name a "DD *" Job Control statement, after which are placed the user's data cards. A "/*" card is placed after the last data card. Some examples of UPDATE runs are:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SYSIN DD *
:UPDATE,FILE=TRAFFIC,FUNCTION=INSERT,DDNAME=ADD
:UPDATE,FILE=TRAFFIC,FUNCTION=NEW-KEY,DDNAME=CHANGE
/*
//ADD DD *
     Data cards for INSERT function
/*
//CHANGE DD *
     Data cards for NEW-KEY function
/*
```

When updating, records are first edited for possible data errors. Two types of errors may be detected: "severe errors" that cause rejection of data cards, and "warnings" that do not. Severe error messages are printed in the format "***** (E) message *****." Warnings are printed in the format "***** (W) message *****." A complete list of error messages is included in Appendix C.

The DELETE Function

The DELETE function is used to delete records from the file. Data cards are prepared in the format:

1-15	Key of record being deleted
16-80	Blank

The INSERT Function

The INSERT function is used to add records to the file. Data cards are prepared as directed in Appendix C. Either an "A" card, a "B" card, or both may be used to insert a record.

The REWRITE Function

This function is used to alter any fields within records other than the key field. Data cards are prepared as directed in Appendix C, coding the key field and any fields being altered. To set a numeric field to zeroes, code zeroes or dollar signs throughout the field. To set a character field to blanks, code dollar signs throughout the field. An "A" card, a "B" card, or both may be submitted when rewriting.

The NEW-KEY Function

The NEW-KEY function is used to alter a record's key field. Data cards are prepared in the format:

1-15	Key of existing record
16	Equal sign
17-31	New key
32-80	Blank

UPDATE-BY-YEAR

This program is run annually to shift the data years one position, clearing the fourth field for filling in with the data during the upcoming year. An example of its use is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SYSIN DD *
:UPDATE-BY-YEAR,FILE=TRAFFIC
/*
```

COPY

COPY is used to generate a backup copy of the file. Its use requires knowledge of OS/VS1 Job Control Language. A SAVETRF DD statement is supplied to allocate a tape or disk output file for use by the COPY program. The DCB parameters LRECL=80 and RECFM=FB are always coded. The user must also supply a blocksize that is a multiple of 80. An example of COPY is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SAVETRF DD UNIT=(TAPE,,DEFER),VOL=SER=545454,
//            DISP=(NEW,KEEP),DSNAME=HIS.TRAFFIC.BACKUP,
//            DCB=(BLKSIZE=8000,LRECL=80,RECFM=FB)
//SYSIN DD *
:COPY,FILE=TRAFFIC
/*
```

CREATE

CREATE is used to reload the file from a previous backup generated by COPY or REORGANIZE. A sample run is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SAVETRF DD UNIT=(TAPE,,DEFER),VOL=SER=545454,
//            DISP=OLD,DSNAME=HIS.TRAFFIC.BACKUP
//SYSIN DD *
:CREATE,FILE=TRAFFIC
/*
```

REORGANIZE

File updates result in wasted storage space, requiring periodic reorganization. Reorganization is performed by generating a backup copy and reloading. This can be accomplished in two steps using COPY and CREATE, or in one using REORGANIZE. A sample run is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SAVETRF DD UNIT=(TAPE,,DEFER),VOL=SER=545454,
//            DISP=(NEW,KEEP),DSNAME=HIS.TRAFFIC.BACKUP,
//            DCB=(BLKSIZE=8000,LRECL=80,RECFM=FB)
//SYSIN DD *
:REORGANIZE,FILE=TRAFFIC
/*
```

The Traffic Report File

The traffic report file is generated from the traffic and true mileage files for producing the Federal Aid Traffic by Sections report and for use by programs in other Highway Information System subsystems. Data elements stored in the report file are:

- Key
- Remark
- Section length
- Data for three years including:
  - Vehicle miles
  - Vehicle miles -- out-of-state vehicles
  - Vehicle miles -- commercial vehicles

Records in the report file are classified as "data" records or as "descriptor" records. One data record is stored for each traffic section. The remark code on a data record is W, T, O, or N to indicate the section type as in the traffic file. A descriptor record with code S, L, or C is stored for each corresponding traffic record. A descriptor record with code D is stored at the end of a section preceding a discontinuity (spur, loop, or coincident section), and a descriptor record with code E is stored at the end of the route. The end-of-route record is followed by a data record with milepoint "999RURAL" containing total vehicle miles for the route. Following the last record of a system is a data record with key "s99999RURAL" with total vehicle miles for the system ("s" is I, P, S, U, or L).

CREATE-TRAFREP

This program generates the traffic report file from the traffic and true mileage files. An example of a CREATE-TRAFREP command with OS/VS1 Job Control language is:

```
// JOB
// EXEC HISTRAF,DISP=OLD
//SYSIN DD *
:CREATE-TRAFREP
/*
```

Optionally, when debugging data, only a portion of the file may be generated by including a DATA parameter:

$$\text{DATA=}\left\{\begin{array}{l}\left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL}\end{array}\quad\left[\text{=n}\ \left[\text{-n}\right]\right]\right\}\\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL}\end{array}\right\}\end{array}\right\}$$

During execution, CREATE-TRAFREP provides some traffic data editing. In the event that two records in the file have remark codes that cannot logically follow each other (such as an M record within a rural section), the message "***** REMARK SEQUENCE ERROR (remark1, remark2) AT KEYS key1 AND key2 *****" is printed. A second error that can arise is a rural record with blank data in the current year position, generating the message "***** INVALID BLANK DATA ON RURAL RECORD key *****."

LIST-TRAFREP

LIST-TRAFREP provides a formatted listing of the report file records specified. The only required parameter is:

$$\text{DATA=}\left\{\begin{array}{l} \left\{\begin{array}{ll} \text{INTERSTATE} & \\ \text{PRIMARY} & \\ \text{SECONDARY} & \left[=\text{n}\left[-\text{n}\right]\right] \\ \text{URBAN} & \\ \text{LOCAL} & \end{array}\right\} \\ \left\{\begin{array}{l} \text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID} \\ \text{ALL} \end{array}\right\} \end{array}\right\}$$

Optional parameters that may be specified are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

A sample run of LIST-TRAFREP follows:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:LIST-TRAFREP,DATA=INTERSTATE
:LIST-TRAFREP,DATA=PRIMARY=1-10
/*
```

The Annual Federal Aid Traffic by Sections Report

The following programs are implemented to produce the annual Traffic by Sections report:

TRAFFIC-BY-SECTIONS	Prints listing of traffic sections
SUMMARY-BY-ROUTES	Prints summary of vehicle miles by routes

TRAFFIC-BY-SECTIONS

This program prints a listing of the traffic major sections, including the milepoint, description, county, section length, AADT for three years, and current vehicle miles. This report forms the main body of the Traffic by Sections report. Required parameters on TRAFFIC-BY-SECTIONS commands are:

REPORT=TRAFFIC

$$\text{DATA=}\left\{\begin{matrix} \left\{\left\{\begin{matrix}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL}\end{matrix}\right.\;\left[\text{=n}\;\left[\text{-n}\right]\right]\right\} \\ \left\{\begin{matrix}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL}\end{matrix}\right\}\end{matrix}\right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Sample commands with associated OS/VS1 Job Control statements are:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=INTERSTATE
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=PRIMARY=4
/*
```

SUMMARY-BY-ROUTES

This program prints a summary of vehicle miles by routes.

Required parameters are:

REPORT=TRAFFIC

$$\text{DATA=} \left\{ \begin{array}{l} \text{INTERSTATE} \\ \text{PRIMARY} \\ \text{SECONDARY} \end{array} \right\}$$

Sample commands with OS/VS1 Job Control statements are:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=PRIMARY
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=INTERSTATE
/*
```

The Annual Federal Aid County Summary

The traffic report file and roadlog file are utilized in producing a summary by county of traffic information. The traffic file must be created prior to running the county summary.

SUM-BY-COUNTY

This program prints the annual county summary. A sample run is:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
:SUM-BY-COUNTY,REPORT=TRAFFIC
/*
```

CHAPTER 5

ACCIDENT SUBSYSTEM

Introduction

Accident data is stored in two separate files: A "detail" file with information containing accident overall details, and a "vehicle" file with information pertaining to vehicles and pedestrians involved in accidents. The accident files are keyed on an assigned accident number rather than the route and milepoint used in other files for two reasons: (1) more than one accident may occur at any given milepoint, and (2) many accidents occur at locations other than may be specified with route and milepoint, such as city streets. A "directory" file is utilized to access on-system accidents by route and milepoint.

A detailed discussion on accident data coding may be found in the publication "A Manual Describing the Proper Use of the State of Montana Investigator's Accident Report."

Accident Record Coding

An accident detail record is coded for each accident entered into the files. This requires either one or two cards. An "A" type card is always used. A "B" card is used for investigated accidents to enter the date and time notified and arrived.

One accident vehicle record is coded for each vehicle and for each pedestrian involved in the accident. Two data cards, a "C" and a "D"

card, are used for entering vehicle records. Occasionally, additional injuries and fatalities occur that cannot be coded within a single vehicle record. In this case, additional "I" cards are submitted with the vehicle record.

Detailed information on coding data cards is contained in Appendix D.

Parameters on Accident Commands

Most file maintenance and report commands within the accident subsystem allow several parameters for selecting accidents. These parameters are described here to save repeating the descriptions for each command.

START-DATE and END-DATE

These parameters are optional on many commands, and are used to select accidents by date. If neither parameter is coded, no selection based on date is done (i.e., all of the records in the file are included). When START-DATE is present, only accidents occurring on or following that date are considered. When END-DATE is present, only accidents occurring on or before the coded date are included. Each date is coded as a two-digit month, day, and year separated by slashes. For example, code November 2, 1974 as 11/02/74.

START-ACCIDENT and END-ACCIDENT

These parameters are used to select accidents based on accident number. If neither is coded, no selection based on accident number is performed. If START-ACCIDENT is present, only accidents having a number equal to or larger than the specified number are included. If END-ACCIDENT is coded, only accidents with a number less than or equal to the specified value are processed. The accident number is always coded as a 12-digit value.

START-ACCIDENT and END-ACCIDENT provide an efficient means of selecting accidents based on reporting agency or on a particular officer during a single year. For example, all accidents coded by Great Falls policemen during the year 1972 begin with the five digits 72052. The parameters START-ACCIDENT=720520000000,END-ACCIDENT=720529999999 cause only 1972 accidents investigated by Great Falls policemen to be processed. The next three digits give an officer's badge number. To process accidents investigated by officer 356 in Great Falls during 1972, code START-ACCIDENT=720523560000, END-ACCIDENT=720523569999.

LOCATION, CITY, and COUNTY

These parameters provide selection by location. Only one of these parameters may be coded on a single command. The LOCATION parameter has two forms: LOCATION=STATEWIDE which processes legally reportable statewide accidents, and LOCATION=EVERYTHING

which processes all statewide accidents. The CITY parameter is coded as CITY=city, where "city" is one of the city names in Table 5-1. The COUNTY parameter is coded as COUNTY=county, where "county" is one of the county names in Table 5-2. In the absence of any of these parameters, LOCATION=STATEWIDE is assumed. When LOCATION=EVERYTHING, CITY=city, or COUNTY=county is specified, accidents are processed regardless of whether they were legally reportable.

SELECT-DD

This parameter provides a means of accident selection based on almost any data element or set of elements within the accident files. Due to the large number of selection options available, it is not feasible to specify the criteria directly on a HIS command. A separate "select statement" is prepared and submitted along with the command. The SELECT-DD parameter provides a linkage from the command to the select statement. The use of SELECT-DD and instructions for coding select statements are included later in this chapter.

Accident File Maintenance

The following programs are implemented in the accident subsystem for maintaining the accident detail and vehicle files:

TABLE 5-1

NAMES CODED IN CITY PARAMETER

ALBERTON	FLAXVILLE	OPHEIM
ANACONDA	FORSYTH	OUTLOOK
BAINVILLE	FORT-BENTON	PHILIPSBURG
BAKER	FROID	PLAINS
BEARCREEK	FROMBERG	PLENTYWOOD
BELGRADE	GERALDINE	PLEVNA
BELT	GLASGOW	POLSON
BIG-SANDY	GLENDIVE	POPLAR
BIG-TIMBER	GRASSRANGE	RED-LODGE
BILLINGS	GREAT-FALLS	REXFORD
BOULDER	HAMILTON	RICHEY
BOZEMAN	HARDIN	RONAN
BRIDGER	HARLEM	ROUNDUP
BROADUS	HARLOWTON	RYEGATE
BROADVIEW	HAVRE	SACO
BROCKTON	HELENA	ST-IGNATIUS
BROWNING	HINGHAM	SCOBEY
BUTTE	HOBSON	SHELBY
CASCADE	HOT-SPRINGS	SHERIDAN
CHESTER	HYSHAM	SIDNEY
CHINOOK	ISMAY	STANFORD
CHOTEAU	JOLIET	STEVENSVILLE
CIRCLE	JORDAN	SUNBURST
CLYDE-PARK	JUDITH-GAP	SUPERIOR
COLUMBIA-FALLS	KALISPELL	TERRY
COLUMBUS	KEVIN	THOMPSON-FALLS
CONRAD	LAUREL	THREE-FORKS
CULBERTSON	LAVINA	TOWNSEND
CUT-BANK	LEWISTOWN	TROY
DARBY	LIBBY	TWIN-BRIDGES
DEER-LODGE	LIMA	VALIER
DENTON	LIVINGSTON	VIRGINIA-CITY
DILLON	LODGE-GRASS	WALKERVILLE
DODSON	MALTA	WESTBY
DRUMMOND	MANHATTAN	WEST-YELLOWSTONE
DUTTON	MEDICINE-LAKE	WHITEFISH
EAST-HELENA	MELSTONE	WHITEHALL
EKALAKA	MILES-CITY	WH-SULPHUR-SPRINGS
ENNIS	MISSOULA	WIBAUX
EUREKA	MOORE	WINIFRED
FAIRFIELD	NASHUA	WINNETT
FAIRVIEW	NEIHART	WOLF-POINT

TABLE 5-2

NAMES CODED IN COUNTY PARAMETER

BEAVERHEAD	GRANITE	POWDER-RIVER
BIG-HORN	HILL	POWELL
BLAINE	JEFFERSON	PRAIRIE
BROADWATER	JUDITH-BASIN	RAVALLI
CARBON	LAKE	RICHLAND
CARTER	LEWIS-AND-CLARK	ROOSEVELT
CASCADE	LIBERTY	ROSEBUD
CHOUTEAU	LINCOLN	SANDERS
CUSTER	MCCONE	SHERIDAN
DANIELS	MADISON	SILVER-BOW
DAWSON	MEACHER	STILLWATER
DEER-LODGE	MINERAL	SWEET-GRASS
FALLON	MISSOULA	TETON
FERGUS	MUSSELSHELL	TOOLE
FLATHEAD	PARK	TREASURE
GALLATIN	PETROLEUM	VALLEY
GARFIELD	PHILLIPS	WHEATLAND
GLACIER	PONDERA	WIBEAUX
GOLDEN-VALLEY		YELLOWSTONE

LIST	Prints selected accidents in abbreviated or formatted listing
UPDATE	Add, rewrites, and deletes accidents
COPY	Generates a backup copy of the files
CREATE	Reloads the files from a backup copy
REORGANIZE	Reformats the files by generating a backup copy and reloading

LIST

LIST provides a listing of the selected accidents. The listing may be an abbreviated listing showing only the accident number, city, county, date, time, and location (LIST=NOT-FORMATTED), or it may be a formatted listing requiring approximately one page per accident (LIST=FORMATTED). Following the listing is a small summary showing the number of accidents, number of injury accidents, number of fatality accidents, number of injuries, and number of fatalities. The only required parameter on accident LIST commands is FILE=ACCIDENT. These parameters are optional:

MAX-#-ACCIDENTS=n

START-ACCIDENT=accident-number

END-ACCIDENT=accident-number

START-DATE=mm/dd/yy

END-DATE=mm/dd/yy

LOCATION= $\left\{ \begin{array}{l} \text{STATEWIDE} \\ \text{EVERYTHING} \end{array} \right\}$

[illegible]

COUNTY=county

SELECT-DD=ddname

LIST= { NOT-FORMATTED / FORMATTED }

LIST=NOT-FORMATTED is assumed if LIST is not coded. The MAX-#-ACCIDENTS parameter may be used to limit the number of accidents listed -- after processing the specified number, the program halts. In the absence of this parameter, no limit is placed on the number of accidents processed. If present, the value coded must range from 1 to 99999.

Some sample LIST commands with accompanying OS/VS1 JCL are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
 LIST,FILE=ACCIDENT,MAX-#-ACCIDENTS=1000,
      START-DATE=01/01/72,END-DATE=01/31/72,
      LIST=NOT-FORMATTED
 LIST,FILE=ACCIDENT,LIST=FORMATTED,
      START-ACCIDENT=720003410305,END-ACCIDENT=720003410305
 LIST,FILE=ACCIDENT,CITY=MISSOULA,START-DATE=06/01/74,
      END-DATE=06/30/74,MAX-#-ACCIDENTS=500
```

UPDATE

UPDATE is used to update the detail and vehicle files. Three transaction types are available, and the user must keep the three data card types separate. A separate command is entered for each transaction type within a given run. Any number of updates within

a transaction type may be submitted with one command. Parameters always coded on UPDATE commands are:

FILE=ACCIDENT

DDNAME=name

$$\text{FUNCTION=} \left\{ \begin{array}{l} \text{DELETE} \\ \text{INSERT} \\ \text{REWRITE} \end{array} \right\}$$

The DDNAME parameter is used to link the user's data cards to the appropriate UPDATE command. Any name of eight characters or less that does not occur within the cataloged procedure HISACC may be coded. If unfamiliar with HISACC, the following names may be safely used:

UPDACCD -- DELETE
UPDACCI -- INSERT
UPDACCR -- REWRITE

The same name coded in the DDNAME parameter is used on a "DD *" Job Control card, after which are placed the user's data cards. A "/*" card is placed after the last data card. Some examples of UPDATE commands with OS/VS1 Job Control statements are:

```
// JOB
// EXEC HISACC,DISP=OLD
[illegible]
:UPDATE,FILE=ACCIDENT,DDNAME=INN,FUNCTION=INSERT
:UPDATE,FILE=ACCIDENT,DDNAME=REWRITE,FUNCTION=REWRITE
[illegible]
[illegible]
     Data cards for INSERT function

//REWRITE DD *
     Data cards for REWRITE function
[illegible]
```

When inserting and rewriting accidents, all transactions are edited for possible data errors. Two levels of editing occurs, resulting in "severe error" messages and "warning" messages. Severe errors are those errors the program cannot possibly accept, and the data cards are rejected. Warnings are possible errors [illegible] update. In many cases, the program attempts to correct the condition causing a warning and the message printed indicates any corrective action taken. A complete list of messages that can be printed by UPDATE may be found in Appendix D.

[illegible]

The DELETE function is used to delete accidents from the [illegible]:

1-12	Key of accident being deleted
13-80	Blank

The INSERT Function

The INSERT function is used to add accidents to the files. Data cards are prepared as directed in Appendix D.

The REWRITE Function

The REWRITE function is used to alter fields within accidents other than the accident number. Data cards are prepared as directed in Appendix D, coding the key and any fields being altered. The combination of cards required to update records depends upon the type of update being done. To rewrite fields from an A card, code "AO" in columns 1-2, the accident number in columns 3-14, and any fields being altered (leave unused columns blank). To rewrite fields from a B card, code "BO" in columns 1-2, the accident number in columns 3-14, and any fields being altered. To rewrite fields from a C card, two data cards are required becaue the C card contains insufficient data to indicate a particular vehicle or pedestrian. Prepare a C card with "CO" in columns 1-2, the accident number in columns 3-14, and any fields being altered. Also prepare a D card with "DO" in columns 1-2, the accident number in columns 3-14, and either the vehicle number in columns 50-51 or the pedestrian number in columns 54-55. Changes on the D card may be coded in columns 15-49, 52-53, and 56-80. To rewrite fields from a D card only, prepare a D card as described

above for rewriting C cards. To rewrite fields from an I card, code "IO" in columns 1-2, the accident number in columns 3-14, and the I card number in columns 50-51. Code any changes in columns 20-49 and 64-78.

COPY

COPY is used to generate a backup copy of the accident files. Use of COPY requires a knowledge of OS/VS1 Job Control Language, as a SAVEACC DD statement must be provided to allocate the backup file on tape or disk. The user need not code DCB parameters, as these are specified within the program. A sample run is:

```
// JOB
// EXEC HISACC,DISP=OLD
//SAVEACC DD UNIT=(TAPE,,DEFER),VOL=SER=444444,
//             DISP=(,KEEP),DSNAME=HIS.ACCIDENT.BACKUP
[illegible]
:COPY,FILE=ACCIDENT
[illegible]
```

CREATE

CREATE is used to reload the accidents from a previous backup copy. A sample run is:

```
// JOB
// EXEC HISACC,DISP=OLD
//SAVEACC DD UNIT=(TAPE,,DEFER),VOL=SER=444444,
//             DISP=OLD,DSNAME=HIS.ACCIDENT.BACKUP
//SYSIN DD *
:CREATE,FILE=ACCIDENT
[illegible]
```

REORGANIZE

Insertion and deletion of accidents results in unused space in the files, and in longer execution times. The files must be periodically reorganized. Reorganization is accomplished by creating a backup copy of the file and reloading. This may be done in two steps using COPY and CREATE, or in one using REORGANIZE. A sample run is:

```
// JOB
// EXEC HISACC,DISP=OLD
//SAVEACC DD UNIT=(TAPE,,DEFER),VOL=SER=444444,
//           DISP=(,KEEP),DSNAME=HIS.ACCIDENT.BACKUP
//SYSIN DD *
:REORGANIZE,FILE=ACCIDENT
/*
```

Accident Memos

One application of the accident files is the printing of accident memos for drivers involved in accidents. Program PRINT-MEMOS scans the accident file for accidents whose memos are to be printed, builds an output file with memo information, sorts this file in order by last name, and prints the memos. If this program completes the sort operation, but fails during the print operation, RESTART-MEMOS may be used to complete the printing.

PRINT-MEMOS

Required parameters on PRINT-MEMOS commands are:

START-DATE=mm/dd/yy

END-DATE=mm/dd/yy

A sample run for printing memos for accidents occurring in April, 1974 is:

```
// JOB
// EXEC HISMEMO,DISP=OLD
//SYSIN DD *
:PRINT-MEMOS,START-DATE=04/01/74,END-DATE=04/30/74
/*
```

RESTART-MEMOS

RESTART-MEMOS continues printing memos after a file has been built and sorted. If the entire memos file is to be printed, no parameters are coded. If some memos have already been printed, code a LOCATION parameter specifying the last name printed. An example of its use is:

```
// JOB
// EXEC HISMEMO
//SYSIN DD *
:RESTART-MEMOS,LOCATION='MCDONALD,RONALD'
//
```

Use of quotes around the name is required if a comma or blank is coded within the name. Quotes are not required if only the last name is specified.

Accident Summaries

The following programs are provided for summarizing the accident detail and vehicle files:

COUNT-ACCIDENTS	Prints number of accidents, number of injury accidents, number of fatality accidents, number of injuries, and number of fatalities
SUM-BY-DAY-&-TIME	Prints summary of accidents by time of day and by day of week
SUM-BY-CONTR-CIRC	Prints summary of accidents involving contributing circumstances
FORM-16	Prints a set of 19 summaries for the National Safety Council form 16 report
SUM-BY-TRAFFICWAY	Prints a set of 41 summaries for the Montana Highway Patrol Bureau's annual report
MOTORCYCLE-SUMMARY	Prints a set of 21 summaries of motorcycle accidents

Each of these programs provides the same set of optional parameters:

START-ACCIDENT=accident-number

END-ACCIDENT=accident-number

START-DATE=mm/dd/yy

END-DATE=mm/ddyy

LOCATION= { STATEWIDE | EVERYTHING }

CITY=city

COUNTY=county

SELECT-DD=ddname

These parameters are described earlier in this chapter.

COUNT-ACCIDENTS

COUNT-ACCIDENTS reads the accidents selected by command parameters, and prints the total number of accidents, number of injury accidents, number of fatality accidents, number of injuries, and number of fatalities. Sample commands and associated OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:COUNT-ACCIDENTS,COUNTY=DEER-LODGE,
:       START-DATE=12/01/73,END-DATE=12/31/73
/*
```

SUM-BY-DAY-TIME

SUM-BY-DAY-&-TIME provides summaries of accidents by day of week and by hour of occurrence. When CITY=city is specified, one summary is printed showing the accidents within that city. When COUNTY=county, LOCATION=STATEWIDE, or LOCATION=EVERYTHING is used,

a rural summary, a municipal summary, and a combined summary are printed. Sample commands with accompanying OS/VS1 JCL are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:SUM-BY-DAY-&-TIME,LOCATION=STATEWIDE,
:    START-DATE=09/01/72,END-DATE=09/30/72
:SUM-BY-DAY-&-TIME,COUNTY=DEER-LODGE
/*
```

SUM-BY-CONTR-CIRC

This program prints summaries of those accidents involving contributing circumstances according to rural-municipal location. Six summaries are printed:

Federal Aid Interstate
Federal Aid Primary
Federal Aid Secondary
Federal Aid Urban
Accidents not on Federal Aid system
All accidents

Some sample SUM-BY-CONTR-CIRC commands with OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:SUM-BY-CONTR-CIRC,CITY=MISSOULA
:SUM-BY-CONTR-CIRC,START-DATE=09/01/72,END-DATE=09/07/72,
    COUNTY=MISSOULA
/*
```

FORM-16

FORM-16 produces 19 of the 21 summaries of the National Safety Council [illegible] summaries are printed for each FORM-16 command. When LOCATION or COUNTY is used, the three sets are:

Rural accidents
Municipal accidents
All accidents

Otherwise, when CITY is used, the sets are:

Legally reportable accidents
Non-reportable accidents
All accidents

Some sample commands and OS/VS1 JCL are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
 FORM-16,START-DATE=06/01/72,END-DATE=06/30/72,
     CITY=GREAT-FALLS
:FORM-16,CITY=GREAT-FALLS,START-DATE=06/01/72,
:    END-DATE=06/30/72,START-ACCIDENT=720523330000,
:    END-ACCIDENT=720523339999
/*
```

SUM-BY-TRAFFICWAY

SUM-BY-TRAFFICWAY prints a set of 41 summaries of accidents, vehicles, drivers, and injuries. Each summary is broken down according to class of trafficway:

Interstate
U. S. highway
State highway
County roads
Local streets

Some sample commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
.SUM-BY-TRAFFICWAY,START-DATE=01/01/72,END-DATE=12/31/72
:SUM-BY-TRAFFICWAY,START-DATE=01/01/72,END-DATE=12/31/72,
:    COUNTY=BROADWATER
```

MOTORCYCLE-SUMMARY

MOTORCYCLE-SUMMARY prints a comprehensive set of summaries of accidents involving motorcycles. Some sample commands with OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
MOTORCYCLE-SUMMARY,START-DATE=01/01/72,END-DATE=03/31/72
MOTORCYCLE-SUMMARY,START-DATE=07/04/74,END-DATE=07/04/74,
    COUNTY=FLATHEAD
/*
```

The Accident Directory File

The accident directory file contains one record for each accident whose location is reported by milepoint. It is used for accessing accidents by milepoint rather than by accident number. For uniqueness, the directory file key consists of the milepoint concatenated with the accident number.

CREATE-FA-ACC-DIREC

This program generates the directory file from the accident detail file. An example of the program's use is:

```
// JOB
// EXEC HISACD,DISP=OLD
//SYSIN DD *
:CREATE-FA-ACC-DIREC
```

The following parameters may be coded on CREATE-FA-ACC-DIREC commands:

```
DATA=  { { INTERSTATE          [ n  [-n] ] } }
       { { PRIMARY                         } }
       { { SECONDARY                       } }
       {                                     }
       { { INTERSTATE+PRIMARY }              }
       { { [illegible]        }              }

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

MAX-#-ACCIDENTS=n
```

DATA, START-MILEPOINT and END-MILEPOINT specify the portion of the directory file being generated. MAX-#-ACCIDENTS specifies the maximum number of records to be written.

LIST-FA-ACC-DIREC

LIST-FA-ACC-DIREC is used to list the directory file. The milepoint location and the accident number is printed from the directory file. The following information is printed from the detail file:

County
Date
Time
Number of vehicles
Number of pedestrians
Number of fatalities
Number of injuries
First harmful event
Roadway-related location
Junction-related location
Weather condition
Road condition
Light condition
Collision type

The following parameter must be coded:

$$\text{DATA=} \left\{ \begin{array}{l} \left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{array}\right. \left[\text{-n} \left[\text{-n}\right]\right] \Bigg\} \\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\end{array}\right\} \end{array}\right\}$$

The following parameters are optional:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

START-DATE=mm/dd/yy

END-DATE=mm/dd/yy

START-ACCIDENT=accident-number

END-ACCIDENT=accident-number

LOCATION= $\left\{ \begin{array}{l} \text{STATEWIDE} \\ \text{EVERYTHING} \end{array} \right\}$

CITY=city

COUNTY=county

SELECT-DD=ddname

Some example commands with accompanying OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:LIST-FA-ACC-DIREC,DATA=INTERSTATE=90
:LIST-FA-ACC-DIREC,DATA=PRIMARY,START-DATE=01/01/73,
:    END-DATE=12/31/73,LOCATION=EVERYTHING
/*
```

The Accident Report File

The accident report file is used in producing the annual accident-by-sections report. The accident directory file is used in generating the report file, and must be generated first.

The report file contains information pertaining to sections of federal aid highway. The sections are identical to those used in producing the traffic-by-sections report, and the traffic file is used to generate the sections. The report file contains data from the accident files, the roadlog file, the traffic file, and the true mileage file.

One "data" record is stored for each traffic section. A remark code of blank, T, O, or N indicates a rural, municipal, out-of-state, or non-existent section. A C, S, or L "descriptor" record is stored at each traffic C, S, or L descriptor. A D precedes each C, S, and L descriptor except those occurring at the beginning of a route. An E descriptor occurs at the end of each route. Data elements in report records are:

- Key (milepoint)
- Remark
- Section description
- Section length
- Data for three years:
  - Year
  - ADT
  - Number of accidents
  - Number of injury accidents
  - Number of fatality accidents
  - Number of injuries
  - Number of fatalities
- Number of lanes
- City number

CREATE-ACC-BY-SECTN

This program generates the accident report file. The command may be coded with no parameters, in which case the complete file is generated. The following parameters, however, are available to aid in debugging data:

$$\text{DATA=}\left\{\begin{matrix}\left\{\begin{matrix}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{matrix}\right.\quad\left[\text{=n}\ \left[\text{-n}\right]\right]\quad\right\}\\ \left\{\begin{matrix}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\end{matrix}\right\}\end{matrix}\right\}$$

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

$$\text{LIST=}\left\{\begin{matrix}\text{ALL}\\ \text{FINAL}\\ \text{ERROR}\end{matrix}\right\}$$

$$\text{CHECKPOINT=}\left\{\begin{matrix}\text{SECTIONS}\\ \text{TRAFFIC}\\ \text{ACCIDENT}\end{matrix}\right\}$$

$$\text{RESTART=}\left\{\begin{matrix}\text{TRAFFIC}\\ \text{ACCIDENT}\\ \text{LOAD}\end{matrix}\right\}$$

DEBUG=n

In the absence of a DATA parameter, INTERSTATE+PRIMARY is used by default.

CREATE-ACC-BY-SECTN operates in four phases: a "sections" phase in which the traffic file is scanned and the report sections

defined; a "traffic" phase in which average daily traffic is computed; an "accident" phase in which accident and roadlog data is retrieved; and a "load" phase in which accident limits are computed and stored in a "limits" file and the report file is stored in its final form.

The LIST parameter can specify a listing of the file during each of the four phases (LIST=ALL), a listing of only the last phase executed (LIST=FINAL), or a listing only in the event of an error (LIST=ERROR). LIST=ERROR is assumed by default in the absence of a LIST parameter.

The CHECKPOINT parameter allows the user to terminate the program at the end of a specified phase. For example, to execute only the sections and traffic phases, code CHECKPOINT=TRAFFIC. The RESTART parameter begins execution at the specified phase, provided that the program has previously executed to at least the phase preceding the phase specified. If the program is abnormally terminated by the Operating System, or if an error occurs and DEBUG=5 was specified, an OS dump will be printed, and it will not be possible to restart execution at any point other than the beginning. If, however, an error occurs that is handled internally (an error message and a termination summary are printed), it is possible to restart at the phase during which the error occurred or at any preceding phase.

DEBUG allows the generation of additional printed output during execution to aid in locating errors. DEBUG=5 indicates all possible additional output and that, in the event of an error, a

user-100 ABEND is generated and an OS dump produced. When this debug level is specified, it is not possible to resume execution with a RESTART parameter in a subsequent run. Other debug levels depend upon the phase executing:

Sections phase:

DEBUG=3, DEBUG=4, or DEBUG=5 results in a listing of the traffic file key and remark.

Traffic phase:

DEBUG not implemented.

Accident phase:

DEBUG=1, DEBUG=2, DEBUG=3, DEBUG=4 or DEBUG=5 requests a listing of all accidents in the directly file not included, with the reason for exclusion.

DEBUG=3, DEBUG=4, or DEBUG=5 requests a listing of all accidents included, and a listing of all roadlog records read.

Load phase:

DEBUG=1, DEBUG=2, DEBUG=3, DEBUG=4, or DEBUG=5 requests a listing of accident limits calculated for each route.

DEBUG=2, DEBUG=3, DEBUG=4, or DEBUG=5 requests a summary of values for each route.

DEBUG=4 or DEBUG=5 requests a listing of the number of sections and of the number of accidents in each section within a route.

A sample run that builds the report file for primary route 1 (note that the DATA parameter is not coded when RESTART is used) is:

```
// JOB
// EXEC HISACCA,DISP=OLD
//SYSIN DD *
:CREATE-ACC-BY-SECTN,DATA=PRIMARY=1,LIST=ERROR,
:    CHECKPOINT=ACCIDENT,DEBUG=3
:CREATE-ACC-BY-SECTN,LIST=ERROR,RESTART=LOAD,DEBUG=1
/*
```

<u>LIST-ACC-BY-SECTN</u>

LIST-ACC-BY-SECTN is used to list the accident report file.

A DATA parameter is required:

$$\text{DATA=} \quad \left\{ \begin{array}{l} \left\{ \left\{\begin{array}{l} \text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{array}\right. \quad \left[=\text{n} \left[-\text{n} \right] \right] \right\} \\ \left\{\begin{array}{l} \text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\end{array}\right\} \end{array} \right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

A sample run, with accompanying OS/VS1 JCL, is:

```
// JOB
// EXEC HISACCA
//SYSIN DD *
:LIST-ACC-BY-SECTN,DATA=PRIMARY=3-6
/*
```

LIST-ACC-LIMITS

LIST-ACC-LIMITS lists the limits file built by CREATE-ACC-BY-SECTN. A DATA parameter is required:

$$\text{DATA=}\quad\left\{\begin{array}{l}\left\{\begin{array}{ll}\text{INTERSTATE} & \\ \text{PRIMARY} & \left[=n\ \left[-n\right]\right] \\ \text{SECONDARY} & \end{array}\right\} \\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID}\end{array}\right\}\end{array}\right\}$$

A sample run, including OS/VS1 JCL, is:

```
// JOB
// EXEC HISACCA
//SYSIN DD *
:LIST-ACC-LIMITS,DATA=PRIMARY
/*
```

The Annual Accident-by-Sections Report

After generating the accident report file, the accident-by-sections report can be produced.

ACCIDENT-BY-SECTIONS

This program prints the accident by sections report. Required parameters are:

```
REPORT=ACCIDENT

        ⎧ ⎧INTERSTATE                    ⎫ ⎫
        ⎪ ⎨PRIMARY      [=n  [-n]]       ⎬ ⎪
DATA=   ⎨ ⎩SECONDARY                     ⎭ ⎬
        ⎪ ⎧INTERSTATE+PRIMARY⎫             ⎪
        ⎩ ⎩FEDERAL-AID       ⎭             ⎭
```

Optional parameters are:

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

A sample run is:

```
// JOB
// EXEC HISACCA
//SYSIN DD *
:ACCIDENT-BY-SECTIONS,REPORT=ACCIDENT,DATA=INTERSTATE
:ACCIDENT-BY-SECTIONS,REPORT=ACCIDENT,DATA=PRIMARY
/*
```

Municipal Accident Analysis

The municipal accident analysis software is designed to aid in locating those intersections having higher numbers of accidents. Any of Montana's incorporated cities may be analyzed.

Accident locations within a city are specified by means of a two-dimensional coordinate system, or "grid." The grid consists of an x-coordinate ranging from 0000 to 2000 and a y-coordinate ranging from

0000 to 1000, overlaid onto a city map. The grid location of each accident occurring within the city is coded and stored in the accident detail file.

The locations and names of intersections within a city are stored within a "grid file."

BUILD-GRID-TABLE

BUILD-GRID-TABLE is used to add or replace the names and locations of intersections within a city. Data cards are prepared in the following format:

1-4	x-coordinate
5-8	y-coordinate
9-48	Intersection name, left justified
49-80	Blank

BUILD-GRID-TABLE is then executed as:

```
// JOB
// EXEC HISACCM,DISP=OLD
//SYSIN DD *
:BUILD-GRID-TABLE,CITY=city,DDNAME=GRIDCARD
/*
//GRIDCARD DD *
     data cards
/*
```

"city" is any of the city names shown above in Table 5-1.

LIST-GRID-TABLE

LIST-GRID-TABLE is used to list the grid file for all cities. A sample run is:

```
// JOB
// EXEC HISACCM
//SYSIN DD *
:LIST-GRID-TABLE
/*
```

HIGH-ACC-INTERSECTNS

This program provides the intersection analysis. A grid file must be stored for the city being analyzed prior to executing HIGH-ACC-INTERSECTNS. Required parameters on HIGH-ACC-INTERSECTNS commands are:

CITY=city

SQUARE-SIZE=n

Optional parameters are:

START-DATE=mm/dd/yy

END-DATE=mm/dd/yy

ACCIDENTS= $\left\{ \begin{array}{c} \text{INTERSECTION} \\ \text{ALL} \end{array} \right\}$

MAX #-ENTRIES=n

#-ACCIDENTS=n

#-INTERSECTIONS=n

The SQUARE-SIZE parameter indicates the size to be used for an intersection. For example, SQUARE-SIZE=10 indicates that each intersection is assumed to be a square whose side is of length 10 in coordinate units, and whose center is the point specified within the grid file.

ACCIDENTS=INTERSECTION limits the analysis to those accidents coded as intersection or as intersection-related. This is the default when no ACCIDENTS parameter is coded.

The MAX-#-ENTRIES parameter specifies the maximum number of entries that are included in the grid file for the specified city. If the grid file contains 500 or fewer entries, this parameter need not be coded. If the grid file contains more than 500 entries when this parameter is not present, or if it contains more than the number specified in this parameter, the program is aborted.

#-ACCIDENTS and #-INTERSECTIONS are mutually exclusive parameters, and indicate the "mode" the program will run. If neither parameter is coded, an INTSECTN DD statement must be supplied indicating actual intersections being examined. If #-ACCIDENTS is coded, any intersections having at least the number of accidents specified are printed. If #-INTERSECTIONS is coded, the number of intersections specified having the largest number of accidents are printed.

The following example shows a run in which the ten intersections with the highest number of accidents are printed:

```
// JOB
// EXEC HISACCM,DISP=OLD
//SYSIN DD *
:HIGH-ACC-INTERSECTNS,CITY=GREAT-FALLS,ACCIDENTS=ALL,
:    SQUARE-SIZE=14,#-INTERSECTIONS=10
/*
```

The following example shows a run in which all intersections having 15 or more accidents are printed:

```
// JOB
// EXEC HISACCM,DISP=OLD
//SYSIN DD *
:HIGH-ACC-INTERSECTNS,CITY=BILLINGS,ACCIDENTS=INTERSECTION,
:    START-DATE=01/01/73,END-DATE=12/31/73,
:    SQUARE-SIZE=19,#-ACCIDENTS=15
/*
```

The following example shows a run in which individual intersections are examined. The names specified must correspond exactly (including blanks) to names within the grid file for that city.

```
// JOB
// EXEC HISACCM,DISP=OLD
//SYSIN DD *
:HIGH-ACC-INTERSECTNS,CITY=GREAT-FALLS,START-DATE=01/01/73,
:    END-DATE=12/31/73,SQUARE-SIZE=14
/*
//INTSECTN DD *
TENTH AVE S & TWENTY FIFTH ST
SMELTER AVE & FIFTEENTH ST
/*
```

Rural Accident Analysis

The rural accident analysis software aids in locating "clusters" of accidents along roadways. Two functions are provided: (1) locating higher accident locations by scanning highways for clusters, and (2) "analyzing" clusters after they are located.

RURAL-ACC-CLUSTERS

RURAL-ACC-CLUSTERS scans rural highways searching for accident clusters. The user indicates what is meant by an accident cluster by specifying a roadway length and a number of accidents. The program scans the specified route or routes, searching for sections of the given length having at least the given number of accidents. A listing of all accidents within the cluster is printed at each cluster detected.

Required parameters on RURAL-ACC-CLUSTERS commands are:

LENGTH=nn.nn

#-ACCIDENTS=n

$$\text{DATA=}\quad\left\{\begin{matrix} \left\{\begin{matrix}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{matrix} \quad \left[\text{=n}\quad [\text{-n}]\right]\right\} \\ \left\{\begin{matrix}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\end{matrix}\right\}\end{matrix}\right\}$$

Optional parameters are:

START-DATE=mm/dd/yy

END-DATE=mm/dd/yy

MAX-#-ACCIDENTS=n

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

The LENGTH parameter defines the roadway length being examined. The length is specified in miles, and must contain leading zeroes and a decimal point to conform to the format nn.nn. For example, a length of one-half mile is coded as 00.50.

The #-ACCIDENTS parameter defines the minimum number of accidents occurring within the specified roadway length to qualify as a cluster. The value may range from 1 to 999.

MAX-#-ACCIDENTS is used to indicate the maximum number of accidents that will occur within any highway section of the length specified in the LENGTH parameter. If a section is encountered having more than this number, the program will abort. If this parameter is not coded, a default of 30 is used.

Some sample commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISACCA
//SYSIN DD *
:RURAL-ACC-CLUSTERS,LENGTH=00.75,#-ACCIDENTS=5,
:     START-DATE=01/01/73,END-DATE=12/31/73,
:     DATA=PRIMARY=3-14
:RURAL-ACC-CLUSTERS,LENGTH=50.00,#-ACCIDENTS=100,
:     DATA=PRIMARY,MAX-#-ACCIDENTS=500,
:     START-DATE=01/01/72,END-DATE=12/31/74
/*
```

RURAL-ACC-ANALYSIS

RURAL-ACC-ANALYSIS is used for a more complete analysis of a cluster than is provided by RURAL-ACC-CLUSTERS. The analysis consists of a summary of all accidents occurring between two specified milepoints on a roadway. The summary is printed in three parts: (1) a plot of the accidents and physical descriptions between the two milepoints on the roadway (scaled to fit on one printed page), (2) the average daily traffic and accident rate between the two milepoints on the roadway, and (3) a summary describing the accident and vehicle details for each accident occurring on the specified section of roadway. The only required parameter on RURAL-ACC-ANALYSIS commands is:

$$\text{DATA=}\left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{array}\quad\left[\text{=n}\right]\right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

START-DATE=mm/dd/yy

END-DATE=mm/dd/yy

A sample run is:

```
// JOB
// EXEC HISACCA
//SYSIN DD *
:RURAL-ACC-ANALYSIS,DATA=PRIMARY=1,
:    START-MILEPOINT=032+0.680,END-MILEPOINT=033+0.900,
:    START-DATE=01/01/73,END-DATE=08/01/73
:RURAL-ACC-ANALYSIS,DATA=PRIMARY=5,END-MILEPOINT=020+0.000
/*
```

Select Statements

Select statements provide a means of selecting accidents for inclusion in reports. The select statements are separate from HIS commands, and are linked to a command through the use of a SELECT-DD parameter coded on the command. The SELECT-DD parameter indicates the name of an OS DD statement used to input the select statement. Any name of eight characters or less may be used, as long as that name does not appear within the cataloged procedure being used. If unsure of the names used, select a name beginning with the characters "SELECT." An example of the use of SELECT-DD is:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:SUM-BY-DAY-&-TIME,SELECT-DD=SMEAR
/*
//SMEAR DD *
     place select statement here
/*
```

Relationships

A select statement is comprised of one or more "relationships" Each relationship consists of a data element name, a comparison symbol, and a comparison constant. A sample relationship is "D.YEAR *EQ* 72," in which D.YEAR is the name of a data element, *EQ* indicates a comparison, and 72 is a comparison constant. The relationship indicates that only accidents occurring in 1972 are to be processed. A complete list of data element names recognized in select statements is shown in Table 5-3. Comparison symbols that may be coded are:

EQ -- Equal	*NE* -- Not equal
GT -- Greater than	*GE* -- Greater than or equal
LT -- Less than	*LE* -- Less than or equal

Complex Relationships

Complex relationships are two or more relationships separated by "*AND*" or by "*OR*." *AND* operations are performed prior to *OR* operations, and hence have higher "priority." If two relationships are separated by *AND*, accidents are included only if both are true. If two relationships are separated by *OR*, accidents are included if either relationship is true.

Coding Select Statements

When coding a select statement, any number of cards may be used and blanks may be used freely to separate individual items. A maximum of 50 relationships may be used within one statement, each separated by *AND* or by *OR* from the others. Some sample select statements are:

1. D.FHE *EQ* 2 *AND* D.#FAT *NE* 0

2. D.FHE *EQ* 2 *AND* D.#FAT *GT* 0 *OR*
   D.FHE *EQ* 2 *AND* D.#INJ *GT* 0

3. D.FHE *NE* 4

Two rules should be noted for coding more efficient select statements. First, when selection is done on elements from both the detail and the vehicle files, try to code the detail file elements first (for example, code "D.FHE *EQ* 4 *AND* V.BODY *EQ* 8" rather than "V.BODY *EQ* 8 *AND* D.FHE *EQ* 4"). Second, note the frequency of occurrence for the relationships whenever possible and order the tests (except where rule 1 is violated) to minimize execution-time testing. For example, if selecting accidents with first harmful event of 4 or 8 and it is known that more accidents exist with 4 than with 8, code "D.FHE *EQ* 4 *OR* D.FHE *EQ* 8" rather than "D.FHE *EQ* 8 *OR* D.FHE *EQ* 4."

Some sample runs with select statements are:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:LIST,FILE-ACCIDENT,SELECT-DD=CNTYFAT
:SUM-BY-TRAFFICWAY,SELECT-DD=SELECTAL
:FORM-16,COUNTY-DATE-LOG6 , SELECT-DD=SELECTDD
/*
//CNTYFAT DD *
  D.#FAT*NE*0 *AND* D.CNTY*GE*2 *AND* D.CNTY*LE* 4
/*
//SELECTAL DD *
  D.CNTY *GE* 2 *AND* D.CNTY *LE* 4 *AND* V.ALC *EQ* 1
/*
//SELECTDD DD *
  V.BODY *EQ* 8
/*
```

TABLE 5-3

DATA ELEMENT NAMES IN SELECT STATEMENTS

Data Element Name	Refers To
------------- Detail Record -- Character Format -------------	
D.KEY	Accident number (12 characters).
D.AGNCY	Accident number -- columns 3-5.
D.BADGE	Accident number -- columns 6-8.
D.LOCN	Location field (12 characters).
D.SYS	Location field -- column 1. Contains: I -- Federal Aid Interstate P -- Federal Aid Primary S -- Federal Aid Secondary U -- Federal Aid Urban R -- Off-system rural M -- Off-system municipal O -- Location not coded
D.ROUTE	Location field -- columns 1-4.
D.PLUS	Location field -- column 8. Contains a plus sign for accidents occurring on rural Federal Aid routes.
D.X	Location field -- columns 5-8. Contains an x-coordinate when D.SYS is "M."
D.Y	Location field -- columns 9-12. Contains a y-coordinate when D.SYS is "M."
D.STUDY	Engineering study. Contains an "X" when an engineering study is requested.
D.REPRT	Legally reportable. Contains an "X" when an accident is legally reportable.
D.INVES	Investigated. Contains an "X" when an accident is not investigated.
------------- Detail Record -- Numeric Format -------------	
D.MONTH	Month of occurrence (1-12).
D.DAY	Day of occurrence (1-31).
D.YEAR	Year of occurrence (72-present year).
D.HOUR	Hour of occurrence (0-24).
D.MIN	Minute of occurrence (0-59).
D.CITY	City number (zero for rural, 1-126 for municipal).
D.CNTY	County number (1-56).
D.FHE	First harmful event (1-11).
D.OBJ	First object hit (0-25).
D.INJ	Injury severity (0-4).
D.DAM	Damage severity (0-3).
D.TRAF	Class of trafficway (1,3,4,6,7, or 8).
D.RDY	Roadway-related location (1-2).

D.JCT	Junction-related location (0-3).
D.#VEH	Number of vehicles (1-99).
D.#PED	Number of pedestrians (0-99).
D.#FAT	Number of fatalities (0-99).
D.#INJ	Number of injuries (0-99).
D.WEATH	Weather condition (0-5).
D.ROAD	Road condition (0-5).
D.LIGHT	Light condition (0-5).
D.CNTRL	Traffic controls (0-16).
D.ODTYP	Other damage type (0-13).
D.ODSEV	Other damage severity (0-3).
D.ODOWN	Other damage owner (0-5).
D.SPEED	Posted speed (0-99).
D.ANAL, D.ANAL1, and D.ANAL2	Analysis fields (0-35). "D.ANAL" refers to both fields, with an implied "*OR*." "D.ANAL1" refers to field 1 and "D.ANAL2" to field 2. Hence, the relationship "D.ANAL *EQ* 2" is equivalent to the relationships "D.ANAL1 *EQ* 2 *OR* D.ANAL2 *EQ* 2."
D.COLL	Collision type (1-7).

------------- Detail Record -- Numeric Computations -------------

C.DAY	Day of Week: 1 Saturday 2 Sunday 3 Monday 4 Tuesday 5 Wednesday 6 Thursday 7 Friday
C.DATE	Date as a six digit number "yymmdd" (year, month,day).
C.TIME	Time as a four digit number "hhmm" (hour, minute).

------------- Vehicle Record -- Character Format -------------

V.TYPE	Record type: "A" Vehicle "B" Pedestrian "C" Additional Injury
V.DRVST	State of driver license (2 characters).
V.EXAM	Re-examination (contains "X" if re-examination is recommended).
V.CHRGE	Charge code (6 characters).
V.HAZRD	Charge code -- Column 6 (contains "H" for hazardous moving violation).
V.SEX and V.SEX1-V.SEX6	Sex codes. "V.SEX" refers to all six positions with an "*OR*" implied.

V.LEV	Damage level (contains "X" for damage in excess of $250).
V.VEHST	Vehicle state of registration (2 characters).

------------- Vehicle Record -- Numeric Format -------------

V.VIS	Vision contributing circumstances (0-6).
V.PHYS	Physical contributing circumstances (0-5).
V.ROAD	Road defects contributing circumstances (0-5).
V.MECH	Mechanical defects contributing circumstances (0-4).
V.VIOL	Possible violations contributing circumstances (0-9).
V.ALC and V.ALC1-V.ALC6	Alcohol. "V.ALC" refers to all six fields, with an implied "*OR*."
V.INJ and V.INJ1-V.INJ6	Injury severity. "V.INJ" refers to all six fields, with an implied "*OR*."
V.AGE and V.AGE1-V.AGE6	Age. "V.AGE" refers to all six fields, with an implied "*OR*."
V.YEAR	Vehicle year (0-99).
V.INTNT	Intent (1-11).
V.BODY	Body style (0-15).
V.TRLR	Trailer style (0-7).
V.DAM	Damage severity (0-3).

NOTE: All vehicle record names have implied "*OR* among all the vehicle records for each accident. Hence, the relationship "V.BODY *EQ* 8" includes all accidents in which any of the vehicle records indicate a body style of 8.

CHAPTER 6

SUFFICIENCY SUBSYSTEM

Introduction

The Montana Department of Highway's Planning and Research Bureau has adopted a sufficiency rating system in order to compare the sufficiency of existing rural highways with the latest design standards. This rating system takes into consideration the structural adequacy, the safety, and the traffic capacity of the highway.

The sufficiency file at present contains rating information only for Montana's Federal Aid Primary route system. The file has been designed, however, to store sufficiency rating information for other systems as the data becomes available.

Sufficiency Record Coding

Sufficiency records contain information about a section of highway rather than about a point on the highway. A sufficiency "section" is identified by specifying the beginning milepoint of the section. Section breaks begin at any point along the highway where a significant change in the highway's structural adequacy, safety, or traffic capacity occurs.

Two major categories of records are stored: "rating" records and "descriptor" records. Descriptor records are identified by special codes in the description field. The card formats for sufficiency records are shown in Appendix E.

Rating records contain the structural and safety ratings for sections of highway. One rating record is stored for each rural sufficiency section.

Six types of descriptor records exist: municipal, coincident, out-of-state, spur, loop and end-of-route. The only fields coded for these records are the key and the description. The description is coded as either CITY, COINCIDENT, OUT-OF-STATE, SPUR, LOOP, or END OF ROUTE.

The Sufficiency Table

A number of constants are required in performing calculations for the sufficiency report. These are stored as five sub-tables in the "sufficiency table." The five sub-tables are:

HV -- Hourly volumes
AF -- Adjustment factors
LF -- Lane factors
SF -- Speed factors
WF -- Width factors

HV consists of 16 records with 11 constants per record. AF consists of 60 records with 5 values each. LF consists of 20 records of 3 constants. SF consists of 3 records with 4 constants. WF consists of 1 record with 5 factors.

LIST-SUFF-TABLE

LIST-SUFF-TABLE prints a listing of the sufficiency table. One line is printed for each record, and each record is identified by one of the codes HV, AF, LF, SF or WF. An example of the use of LIST-SUFF-TABLE is:

```
// JOB
// EXEC HIS
//SYSIN DD *
:LIST-SUFF-TABLE
/*
```

UPDATE-SUFF-TABLE

UPDATE-SUFF-TABLE can rewrite records in the sufficiency table. Data cards are prepared with a record identification in columns 1-5 (the identification can be obtained from a table listing), and up to 11 4-digit values in columns 6-49. An example of the use of UPDATE-SUFF-TABLE is:

```
// JOB
// EXEC HIS,DISP=OLD
//SYSIN DD *
:UPDATE-SUFF-TABLE,DDNAME=UPDSUFF
/*
//UPDSUFF DD *
     data cards
/*
```

Sufficiency File Maintenance

The following programs are implemented for maintaining the sufficiency file:

LIST	Prints records in tabular format
UPDATE	Deletes, inserts, and rewrites records
COPY	Generates backup copy of the file
CREATE	Reloads the file from a backup copy
REORGANIZE	Reformats the file by generating a backup copy and reloading

LIST

LIST prints a tabular listing of selected records in the file. Required parameters are:

FILE=SUFFICIENCY

$$\text{DATA=}\left\{\begin{array}{l} \left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{array} \left[\text{=n}\left[\text{-n}\right]\right]\right\} \\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\end{array}\right\}\end{array}\right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Some sample commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISSUFF
//SYSIN DD *
:LIST,FILE=SUFFICIENCY,DATA=PRIMARY
:LIST,FILE=SUFFICIENCY,DATA=PRIMARY=1,
:      START-MILEPOINT=010+0.000,END-MILEPOINT=060+0.000
/*
```

UPDATE

UPDATE is used to update the file. Four transaction types are available, and the user must keep the four data card types separate. A separate command is entered for each transaction type within a given run. Any number of updates within a transaction type may be submitted with one command. Parameters that must always be coded on an UPDATE command are:

```
FILE=SUFFICIENCY

DDNAME=name

            ⎧ DELETE  ⎫
FUNCTION    ⎨ INSERT  ⎬
            ⎪ REWRITE ⎪
            ⎩ NEW-KEY ⎭
```

The DDNAME parameter links the user's data cards to the UPDATE command. Any name of eight characters or less that does not occur within the cataloged procedure HISSUFF may be coded. If unfamiliar with the contents of HISSUFF, the following names can be safely used:

```
UPDSUFD -- DELETE
UPDSUFI -- INSERT
UPDSUFR -- REWRITE
UPDSUFN -- NEW-KEY
```

The same name is used to name a "DD *" Job Control statement, after which are placed the data cards. A "/*" card is placed after the last data card. Some examples of UPDATE runs are:

```
// JOB
// EXEC HISSUFF,DISP=OLD
//SYSIN DD *
:UPDATE,FILE=SUFFICIENCY,FUNCTION=INSERT,DDNAME=NEWRECS
:UPDATE,FILE=SUFFICIENCY,FUNCTION=REWRITE,DDNAME=SMACK
/*
//NEWRECS DD *
     Data cards for INSERT function
/*
//SMACK DD *
     Data cards for REWRITE function
/*
```

When updating, records are first edited for possible data errors. Two types of errors may be detected: "severe errors" that cause rejection of data cards, and "warnings" that do not. Severe error messages are printed in the format "***** (E) message *****." Warnings are printed in the format "***** (W) message *****." A complete list of error messages is included in Appendix E.

The DELETE Function

The DELETE function is used to delete records from the file. Data cards are prepared in the format:

1-15 -- Key of record being deleted
16-80 -- Blank

The INSERT Function

The INSERT function is used to add records to the file. Data cards are prepared as directed in Appendix E.

The REWRITE Function

This function is used to alter any fields within existing records other than the key field. Data cards are prepared as directed in Appendix E, coding the key and any fields being altered. To set a numeric field to zeroes, code zeroes or dollar signs throughout the field. To set a character field to blanks, code dollar signs throughout the field.

The NEW-KEY Function

The NEW-KEY function is used to alter a record's key field. Data cards are prepared in the format:

1-15 -- Key of existing record
16 -- Equal sign
17-31 -- New key
32-80 -- Blank

COPY

COPY is used to generate a backup copy of the file. Its use requires knowledge of OS/VS1 Job Control Language. A SAVESUF DD statement is supplied to allocate a tape or disk output file for use by the COPY program. The DCB parameters LRECL=68 and RECFM=FB are always coded. The user must also supply a blocksize that is a multiple of 68. An example of COPY is:

```
// JOB
// EXEC HISSUFF,DISP=OLD
//SAVESUF DD UNIT=(TAPE,,DEFER),VOL=SER=999999,
//           DISP=(NEW,KEEP),DSNAME=HIS.SUFF.BACKUP,
//           DCB=(BLKSIZE=6800,LRECL=68,RECFM=FB)
//SYSIN DD *
:COPY,FILE=SUFFICIENCY
/*
```

CREATE

CREATE reloads the file from a backup copy. A sample run is:

```
// JOB
// EXEC HISSUFF,DISP=OLD
//SAVESUF DD UNIT=TAPE,VOL=SER=999999,DISP=OLD,
//           DSNAME=HIS.SUFF.BACKUP
//SYSIN DD *
:CREATE,FILE=SUFFICIENCY
/*
```

REORGANIZE

File updates result in wasted storage space, requiring periodic reorganization. Reorganization is performed by generating a backup copy and reloading. This can be accomplished in two steps using COPY and CREATE, or in one using REORGANIZE. A sample run is:

```
// JOB
// EXEC HISSUFF,DISP=OLD
//SAVESUF DD UNIT=TAPE,VOL=SER=999999,DISP=(NEW,KEEP),
//           DSNAME=HIS.SUFF.BACKUP,
//           DCB=(BLKSIZE=13600,LRECL=68,RECFM=FB)
//SYSIN DD *
:REORGANIZE,FILE=SUFFICIENCY
/*
```

The Sufficiency Report File

The sufficiency report file is generated from files in the roadlog, traffic, true mileage, accident, and sufficiency subsystems. It is used for producing the annual sufficiency-by-sections report. The file contains one record for each record in the sufficiency file. Data elements stored in the report file are:

Key
Remark
Description
County number
Financial district
Year built
Year improved
Surface width
Roadway width
Surface type
Section length

Average daily traffic
Design hour volume
Percent of commercial traffic
Service volume
Number of accidents
Foundation rating
Surface rating
Drainage rating
Safety rating
Capacity rating
Total rating
Adjusted rating
Deficient mileage
Design speed
Terrain
Average speed
Sight distance
Stopping distance
Number of curves
Number of bridges
Number of lanes
Divided/undivided code
City number
Current year traffic

CREATE-SUFFREP

CREATE-SUFFREP generates the sufficiency report file. To create the entire file, no parameters are required on the command. The following parameters are available to aid in data debugging:

$$\text{DATA=}\left\{\begin{array}{l}\left\{\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\end{array}\quad\left[\text{=n}\ \left[\text{-n}\right]\right]\right\} \\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\end{array}\right\}\end{array}\right\}$$

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

LIST { ALL | FINAL | ERROR }

CHECKPOINT= { SUFFICIENCY | ROADLOG | TRAFFIC | ACCIDENT }

RESTART= { ROADLOG | TRAFFIC | ACCIDENT | CALCULATIONS }

DEBUG=5

The DATA, START-MILEPOINT and END-MILEPOINT parameters may be used to limit the portion of the file generated. These parameters are ignored if a RESTART parameter is also present.

The LIST parameter allows a listing of the file to be printed under various conditions. LIST=ERROR, the default value, prints a file listing only if an error occurs. LIST=ALL prints a file listing during each program phase. LIST=FINAL prints a file listing during the final program phase.

The program operates in six phases: sufficiency, roadlog, traffic, accident, calculations, and load. The CHECKPOINT parameter is used to instruct the program to terminate after the specified phase, first invoking the load phase to store intermediate values. The RESTART parameter is used to restart at the specified phase.

DEBUG=5 is normally utilized only by programmers involved in program maintenance. When DEBUG=5 is specified, the normal error-handling procedures are bypassed, and a user-100 ABEND is forced in the event of an error. After an error occurs with DEBUG=5 specified, it is not possible to restart using the RESTART parameter.

Sample CREATE-SUFFREP runs are:

```
// JOB
// EXEC HISSUFF,DISP=OLD
//SYSIN DD *
:CREATE-SUFFREP
/*

// JOB
// EXEC HISSUFF,DISP=OLD
//SYSIN DD *
:CREATE-SUFFREP,DATA=PRIMARY=8,LIST=ALL,
:    CHECKPOINT=TRAFFIC
:CREATE-SUFFREP,LIST=ERROR,RESTART=ACCIDENT
/*
```

The Sufficiency Phase

This phase reads the sufficiency file and prepares a "skeleton" report file. Data elements stored by the sufficiency phase are:

- Key
- Remark
- Description
- Design speed
- Terrain
- Average highway speed
- Percent of sight distances less than 1500 feet
- Number of stopping sight distances less than design
- Number of horizontal curves sharper than the design degree of curvature
- Number of narrow bridges
- Foundation rating
- Surface rating
- Drainage rating
- Section length

The remark is a one-character code indicating the record type. A blank is used to indicate normal rating records. Other codes are:

M -- municipal
C -- coincident
S -- Spur or loop
O -- out of state
E -- end of route
U -- under construction
N -- non-existent

The section length is computed via the true mileage file. All other fields are simply copied from the sufficiency file.

The Roadlog Phase

The roadlog phase simultaneously reads the output file from the sufficiency phase and the roadlog file, copying into the report file the following fields:

Year built
Year improved
Surface width
Roadway width
Surface type
Number of lanes
Divided/undivided code
City number
County number

The Traffic Phase

The traffic phase reads the output file from the roadlog phase, filling in traffic data. Data elements stored are:

Average annual daily traffic (three year average)
Average annual daily traffic (current year)
Design hourly volume
Percent of commercial traffic

Average annual daily traffic is computed for each of three years by summing the vehicle miles and dividing by the section length. The three year average and the current year traffic are saved. The value stored for design hourly volume is computed by first searching the sufficiency section for the traffic sub-section having the highest DHV, and then multiplying this DHV times the current year AADT for the sufficiency section. The value stored for percent commercial traffic is the value coded for percent commercial traffic on the traffic record from which the DHV was taken.

The Accident Phase

The accident phase reads the traffic phase output simultaneously with the accident directory file, and computes the number of accidents occurring in each rural sufficiency section.

The Calculations Phase

The calculations phase reads the accident phase output, and computes the following items from elements already stored in the report file:

- Service volume
- Capacity rating
- Safety rating
- Total rating
- Deficient mileage
- Adjusted rating

Service Volume Calculations -- Four-Lane Roads

For 4-lane roads, the service volume is given by the formula "8000xy." "x" is given by the formula:

$$\frac{100}{100 + a(2^{b}-1)}$$

where "a" is the percent of commercial traffic and "b" is the terrain code. "y" is a factor extracted from the sufficiency tables, utilizing the speed factor and width factor sub-tables. The speed factor is determined from the average speed and the design speed. The width factor is determined from the shoulder width. "y" is the product of the speed factor and the width factor.

Service Volume Calculation -- Two- and Three-Lane Roads

For 2- and 3-lane roads, the service volume is the product of two factors. For 3-lane roads, the first factor is extracted from the lane factor sub-table using the lane width, shoulder width, and terrain. For 2-lane roads, the first factor is extracted from the adjustment factor sub-table using the terrain code, percent commercial, lane width, and shoulder width. The second factor is extracted from the hourly volume sub-table.

Capacity Rating Calculations

The capacity rating is given by the formula:

$$30 - \frac{15a}{b}$$

where "a" is the design hourly volume and "b" is the service volume.

Safety Rating Calculations

The safety rating is a function of hazardous roadway conditions and accident rate. The formula used is:

$$\frac{2x}{N_1+N_2+N_3+N_4}$$

where "x" is the section length, "N_1" is the number of stopping sight distances less than that permitted by design, "N_2" is the number of horizontal curves sharper than the design speed permits, "N_3" is the number of narrow bridges, and "N_4" is the accident rate computed by the formula:

$$\frac{N \times 10^7}{AADT \times 365 \times 100}$$

where "N" is the number of accidents and "AADT" is the 3-year AADT.

Total Rating Calculations

The total rating is the summation of the foundation, surface, drainage, safety, and capacity ratings. The maximum rating is 100.

Deficient Mileage Calculations

The deficient mileage is given by:

$$\frac{x(100-r)}{100}$$

where "x" is the section length and "r" is the total rating.

Adjusted Rating Calculations

The adjusted rating gives priority to those sections of highway carrying larger traffic volumes. The formula for this rating is:

$$x + \frac{x^2 - 100x}{50 \log y} (\log z - \log y)$$

where "x" is the total rating, "y" is the total system current-year AADT, and "z" is the total section current-year AADT.

LIST-SUFFREP

LIST-SUFFREP prints the contents of the sufficient report file. A DATA parameter is required:

$$\text{DATA=} \left\{ \begin{array}{l} \left\{ \begin{array}{ll} \text{INTERSTATE} & \\ \text{PRIMARY} & \left[\text{=n} \; \left[\text{-n} \right] \right] \\ \text{SECONDARY} & \end{array} \right\} \\ \left\{ \begin{array}{l} \text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID} \end{array} \right\} \end{array} \right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Sample commands with OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISSUFF
//SYSIN DD *
:LIST-SUFFREP,DATA=PRIMARY=8
:LIST-SUFFREP,DATA=PRIMARY=4-5
/*
```

The Annual Sufficiency Report

The following programs are used in producing the annual sufficiency report:

LIST-BY-SECTION	Prints listing of report file by section
LIST-BY-DISTRICT	Prints listing of report file by financial district
LIST-BY-RATING	Prints listing of report file by adjusted rating
MAP-TABLES	Prints listing of report file in a table format
RATING-BY-DISTRICT	Prints a table of financial districts showing the percentage of sufficient mileage
DEF-MILES-BY-COUNTY	Summarizes the amount of deficient mileage within each county

Each command requires the parameters:

REPORT=SUFFICIENCY

$$\text{DATA=} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{INTERSTATE} \\ \text{PRIMARY} \\ \text{SECONDARY} \end{array} \left[\text{=n} \left[\text{-n} \right] \right] \right\} \\ \left\{ \begin{array}{l} \text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID} \end{array} \right\} \end{array} \right\}$$

The following parameters are optional:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Some sample runs showing OS/VS1 Job Control Language are:

```
// JOB
// EXEC HISSUFF
//SYSIN DD *
:LIST-BY-SECTION,REPORT=SUFFICIENCY,DATA=PRIMARY=8
:LIST-BY-RATING,REPORT=SUFFICIENCY,DATA=PRIMARY=8
:DEF-MILES-BY-COUNTY,REPORT=SUFFICIENCY,DATA=PRIMARY=8
/*
```

CHAPTER 7

BRIDGE SUBSYSTEM

Introduction

The bridge inventory file has been added to the Highway Information System for release 3.0. One record is stored in this file for each bridge in the state, and contains descriptive and appraisal data.

Bridge Record Coding

The bridge file key consists of the usual 15-character HIS key (route system, route number, and milepoint) plus a one-digit sequence number. The sequence number is necessary because more than one bridge can be located at the same milepoint. The sequence number is 1 for the first or only bridge, 2 for the next, and so on.

When updating the file, a series of five cards are coded for each record. Complete information on preparing these cards is contained in Appendix F.

Defense Section Cross-Reference File

The bridge defense and pre-attack listings require the use of a cross-reference technique to access bridges by defense section number. One data card is prepared for each defense section:

```
 1    -- Federal aid route system
 3-6  -- Federal aid route number
 7-15 -- Starting milepoint of defense section
16-24 -- Ending milepoint of defense section
25-29 -- Defense section number
30-38 -- Inventory route number (coded as in bridge file)
39-43 -- Latitude (coded as in bridge file)
44-48 -- Longitude (coded as in bridge file)
```

The inventory route number is required on all cards. The latitude and longitude needs to be coded only on sections containing no bridges, and is the approximate latitude and longitude of the midpoint of the section.

The data cards are loaded into the computer with the following control cards:

```
// JOB
// EXEC HISBRID
//SYSIN DD *
:SEQLOAD,DDNAME=DEFENSE
/*
//SEQLOAD DD *
     data cards in any order
/*
```

Bridge File Maintenance

The following programs are implemented for maintaining the bridge file:

LIST	Prints records in tabular format
UPDATE	Deletes, inserts, and rewrites records
COPY	Generates backup copy of the file
CREATE	Reloads the file from a backup copy
REORGANIZE	Reformats the file by generating a backup copy and reloading

LIST

LIST prints a tabular listing of selected records in the file. Because the record length prohibits printing all the fields within a single format, several different listing formats are available. Each format prints a different set of fields. Between the six formats available, all of the bridge fields can be printed. Required parameters are:

```
FILE=BRIDGE

        1
        2
        3
LIST=   4
        5
        6

         INTERSTATE
         PRIMARY
         SECONDARY        [=n [-n]]
         URBAN
DATA=    LOCAL

         INTERSTATE+PRIMARY
         FEDERAL-AID
         ALL
```

Optional parameters are:

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

Some sample commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISBRID
//SYSIN DD *
:LIST,FILE=BRIDGE,LIST=1,DATA=PRIMARY=8-10
:LIST,FILE=BRIDGE,LIST=2,DATA=PRIMARY=8-10
/*
```

Fields printed for LIST=1 are:

Key
Remark
Coincident key
Maintenance district
Construction district
Detour length
Features intersected
Minimum vertical clearance
Total horizontal clearance
Major or minor
Latitude
Longitude

Fields printed for LIST=2 are:

Key
Remark
Coincident key
Inventory route
Facility carried
Physical vulnerability
Custodian
Year built
Year improved
Lanes on structure
Lanes under structure
Design load
Bridge median
Skew
Structure flare
Navigation control

Vertical navigation clearance
Horizontal navigation clearance

Fields printed for LIST=3 are:

Key
Remark
Coincident key
Service type
Main structure type
Approach structure type
Number of main spans
Number of approach spans
Maximum span length
Structure length
Left sidewalk width
Right sidewalk width
Bridge roadway width

Fields printed for LIST=4 are:

Key
Remark
Coincident key
Minimum vertical overclearance
Minimum vertical underclearance
Minimum lateral underclearance
Wearing surface
Surface depth
Approach guardrail
Main guardrail
Posted speed limit
Posted load limit
Deck condition
Superstructure
Substructure
Channel and channel protection
Culvert and retaining walls
Remaining life
Operating rating
Approach alignment

Fields printed for LIST=5 are:

Key
Remark
Coincident key
Inventory rating
Structural condition
Deck geometry
Underclearances
Safe load capacity
Waterway adequacy
Approach alignment rating
Year of needed improvement
Type of service
Type of work
Improvement length
Proposed design load
Proposed road width
Proposed number of lanes
Design ADT
Year of design ADT

Fields printed for LIST=6 are:

Key
Remark
Coincident key
Inspection date
Structure batch number
Microfilm serial number
Date of update

UPDATE

UPDATE is used to update the file. Three transaction types are available, and the user must keep the three data card types separate. A separate command is entered for each transaction type within a given run. Any number of updates within a transaction

type may be submitted with one command. Parameters that must always be coded on an UPDATE command are:

FILE=BRIDGE

DDNAME=name

FUNCTION= { DELETE | INSERT | REWRITE | NEW-KEY }

The DDNAME parameter links the user's data cards to the UPDATE command. Any name of eight characters or less that does not occur within the cataloged procedure HISBRID may be coded. If unfamiliar with the contents of HISBRID, the following names can be safely used:

```
UPDBDGD -- DELETE
UPDBDGI -- INSERT
UPDBDGR -- REWRITE
UPDBDGN -- NEW-KEY
```

The same name is used to name a "DD *" Job Control statement, after which are placed the data cards. A "/*" card is placed after the last data card. Some examples of UPDATE runs are:

```
// JOB
// EXEC HISBRID,DISP=OLD
//SYSIN DD *
:UPDATE,FILE=BRIDGE,FUNCTION=INSERT,DDNAME=ADDEM
:UPDATE,FILE=BRIDGE,FUNCTION=DELETE,DDNAME=SUBEM
/*
//ADDEM DD *
     Data cards for INSERT function
/*
//SUBEM DD *
     Data cards of DELETE function
/*
```

When updating, records are first edited for possible data errors. Two types of errors may be detected: "severe errors" that cause rejection of data cards, and "warnings" that do not. Severe error messages are printed in the format "*** ERROR-message." Warnings are printed in the format "*** WARNING - message." A complete list of error messages is included in Appendix F.

The DELETE Function

The DELETE function is used to delete records from the file. Data cards are prepared in the format:

1-16 -- Key of record being deleted
17-80 -- Blank

The INSERT Function

The INSERT function is used to add records to the file. Data cards are prepared as directed in Appendix F.

The REWRITE Function

The REWRITE function is used to alter any fields within existing records other than the key field. Data cards are prepared as directed in Appendix F. Any combination of the five data cards may be used to rewrite a record. On each card included, code the key and any fields being altered. To

replace a character field with blanks, code dollar signs throughout the field. To replace a numeric field with a zero value, code zeroes or dollar signs throughout the field.

COPY

COPY is used to generate a backup copy of the file. Its use requires knowledge of OS/VS1 Job Control Language. A SAVEBDG DD statement is supplied, allocating a tape or disk output file for use by COPY. The DCB parameters LRECL=268 and RECFM=FB are always coded. A blocksize that is a multiple of 268 must also be supplied. An example run is:

```
// JOB
// EXEC HISBRID,DISP=OLD
//SAVEBDG DD UNIT=TAPE,VOL=SER=888888,
//            DISP=(NEW,KEEP),DSNAME=HIS.BRIDGE.BACKUP,
//            DCB=(BLKSIZE=26800,LRECL=268,RECFM=FB)
//SYSIN DD *
:COPY,FILE=BRIDGE
/*
```

CREATE

CREATE reloads the file from a backup copy. A sample run is:

```
// JOB
// EXEC HISBRID,DISP=OLD
//SAVEBDG DD UNIT=TAPE,VOL=SER=888888,DISP=OLD,
//            DSNAME=HIS.BRIDGE.BACKUP
//SYSIN DD *
:CREATE,FILE=BRIDGE
/*
```

REORGANIZE

File updates result in wasted storage space, and periodic reorganization is necessary. Reorganization is performed by generating a backup copy and reloading. This can be accomplished in two steps using COPY and CREATE, or in one using REORGANIZE. A sample run is:

```
// JOB
// EXEC HISBRID,DISP=OLD
//SAVEBDG DD UNIT=TAPE,VOL=SER=888888,
//           DISP=(NEW,KEEP),DSNAME=HIS.BRIDGE.BACKUP,
//           DCB=(BLKSIZE=26800,LRECL=268,RECFM=FB)
//SYSIN DD *
:REORGANIZE,FILE=BRIDGE
/*
```

Bridge Inspection Data Tabulations

The bridge inspection data tabulation is a tape containing data collected under the national bridge inspection programs. The tape is prepared from the bridge, roadlog, and traffic files by the program BDG-INSPECTION-TAPE.

BDG-INSPECTION-TAPE

The only required parameter is:

```
        ┌ ┌ INTERSTATE                             ┐ ┐
        │ │ PRIMARY                                │ │
        │ │ SECONDARY      [=n [-n]]               │ │
DATA=   │ │ URBAN                                  │ │
        │ └ LOCAL                                  ┘ │
        │ ┌ INTERSTATE+PRIMARY ┐                     │
        │ │ FEDERAL-AID        │                     │
        └ └ ALL                ┘                     ┘
```

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Use of BDG-INSPECTION-TAPE requires knowledge of OS/VS1 JCL for allocating tape files. A BDGTAPE DD statement must be supplied for this purpose. A sample run of BDG-INSPECTION-TAPE is:

```
// JOB
// EXEC HISBRID
//BDGTAPE DD UNIT=TAPE,VOL=SER=999999,DSN=HIS.BDGINSP,
//   DISP=(NEW,KEEP),DCB=(BLKSIZE=7200,LRECL=80,RECFM=FB)
//SYSIN DD *
:BDG-INSPECTION-TAPE,DATA=PRIMARY
/*
```

The Planning and Research Bridge Listings

The annual Planning and Research Bridge Listing is produced by the program BDG-INVENTORY-LIST.

The only required parameter is:

$$\text{DATA=} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{INTERSTATE} \\ \text{PRIMARY} \\ \text{SECONDARY} \\ \text{URBAN} \\ \text{LOCAL} \end{array} \quad \left[\text{=n} \; \left[\text{-n} \right] \right] \right\} \\ \left\{ \begin{array}{l} \text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID} \\ \text{ALL} \end{array} \right\} \end{array} \right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

A sample run is:

```
// JOB
// EXEC HISBRID
//SYSIN DD *
:BDG-INVENTORY-LIST,DATA=FEDERAL-AID
/*
```

The Bridge Report File

The bridge report file is used for producing reports about highway defense bridges. The file is generated from the bridge, roadlog, traffic, and true mileage files.

CREATE-BDGREP

CREATE-BDGREP creates the bridge report file. No parameters are required. A sample run is:

```
// JOB
// EXEC HISBRID,DISP=OLD
//SYSIN DD *
:CREATE-BDGREP
/*
```

Highway Defense Bridges

Several reports of highway defense bridges are prepared annually for the Office of Emergency Preparedness (OEP), the Department of Defense (DOD), and the Federal Highway Administrations (FHWA). These reports are obtained from the bridge report file. Programs implemented for producing these reports are:

DEFENSE-BDG-LIST	Prints listing of defense bridges
PRE-ATTACK-BDG-TAPE	Produces a tape with pre-attack information
SUM-BY-DESIGN-LOAD	Prints summary of defense bridges by design load

No parameters are required for these commands. A PREATACK DD statement must be supplied to allocate a tape file for PRE-ATTACK-BDG-TAPE.

Sample runs are:

```
// JOB
// EXEC HISBRID
//PREATACK DD UNIT=TAPE,VOL=SER=999999,
//             DSN=HIS.PATAK,DISP=(NEW,KEEP),
//             DCB=(RECFM=FB,LRECL=50,BLKSIZE=10000)
//SYSIN DD *
:DEFENSE-BDG-LIST
:PRE-ATTACK-BDG-TAPE
:SUM-BY-DESIGN-LOAD
/*
```

CHAPTER 8

RAILROAD SUBSYSTEM

Introduction

The railroad inventory file has been added to the Highway Information System for release 3.0. One record is stored in this file for each railroad crossing in the state. This record contains descriptive and classificatory information on the crossing.

Railroad Record Coding

Each record in the railroad file can be accessed by a fifteen character key (route system, route number, and milepoint). When coding railroad records, this key identifies a particular crossing. Complete information on preparing cards is contained in Appendix G.

Railroad File Maintenance

The following programs are implemented for maintaining the railroad file:

LIST	Prints records in tabular format
UPDATE	Deletes, inserts, and rewrites records
COPY	Generates backup copy of the file

CREATE	Reloads the file from a backup copy
REORGANIZE	Reformats the file by generating a backup copy and reloading

LIST

LIST prints a tabular listing of selected records in the file. Because the record length prohibits printing all the fields within a single format, several different listing formats are available. Each format prints a different set of fields. These fields correspond to those of the three cards used to code a record. Between the three formats available, all of the railroad fields can be listed. Required parameters are:

FILE=RAILROAD

$$\text{LIST=}\left\{\begin{array}{c}\text{A}\\ \text{B}\\ \text{C}\end{array}\right\}$$

$$\text{DATA=}\left\{\begin{array}{l}\left\{\begin{array}{ll}\begin{array}{l}\text{INTERSTATE}\\ \text{PRIMARY}\\ \text{SECONDARY}\\ \text{URBAN}\\ \text{LOCAL}\end{array} & \left[\text{=n}\ \left[\text{-n}\right]\right]\end{array}\right\}\\ \left\{\begin{array}{l}\text{INTERSTATE+PRIMARY}\\ \text{FEDERAL-AID}\\ \text{ALL}\end{array}\right\}\end{array}\right\}$$

Optional parameters are:

START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx

Some sample commands with OS/VS1 JCL are:

```
// JOB
// EXEC HISRRX
//SYSIN DD *
:LIST,FILE=RAILROAD,LIST=A,DATA=SECONDARY
:LIST,FILE=RAILROAD,LIST=C,DATA=PRIMARY=8-10
/*
```

Fields printed for LIST=A are:

Card code
Key
Date of Field Survey
Road Width
Road Approach One
Road Approach Two

Fields printed for LIST=B are:

Card code
Key
Railroad Traffic
Urban/Rural Code
Verbal Description of Crossing Location

Fields printed for LIST=C are:

Card code
Key
Date of Crossing Inventory Form
Operating Railroad Co.
Crossing ID Number
Branch or Line Name
Branch or Line Milepost
Number of Tracks
Use of Tracks by Other Than Operating Railroad

Protection
Number of Traffic Lanes
Nightly Train Traffic
Estimated ADT

UPDATE

UPDATE is used to update the file. Four transaction types are available, and the user must keep the four card types separate. A separate command is entered for each transaction type within a given run. Any number of updates of a specific transaction type may be submitted with each command specifying that particular type of update. Parameters that must be coded on an update command are:

FILE=RAILROAD

DDNAME=name

$$\text{FUNCTION=} \left\{ \begin{array}{l} \text{DELETE} \\ \text{INSERT} \\ \text{REWRITE} \\ \text{NEW-KEY} \end{array} \right\}$$

The DDNAME parameter links the user's data cards to the UPDATE command. Any name of eight characters or less that doesn't occur within the cataloged procedure HISRRX may be coded. If unfamiliar with the contents of HISRRX, the following names can be safely used:

UPDRRXD -- DELETE
UPDRRXI -- INSERT
UPDRRXR -- REWRITE
UPDRRXN -- NEW-KEY

The same name is used to name a "DD *" JCL statement, after which are placed the data cards of a particular transaction type. A "/*" card is placed after the last data card. Some examples of UPDATE are:

```
// JOB
// EXEC HISRRX,DISP=OLD
//SYSIN DD *
:UPDATE,FILE=RAILROAD,FUNCTION=INSERT,DDNAME=ADD
:UPDATE,FILE=RAILROAD,FUNCTION=DELETE,DDNAME=ABC
/*
//ADD DD *
     Data cards for INSERT function
/*
//ABC DD *
     Data cards for DELETE function
/*
```

When updating, records are first edited for possible data errors. Two types of errors may be detected: "severe errors" that cause rejection of data cards, and "warnings" that do not. Severe error messages are printed in the format "***** (E) message." Warnings are printed in the format "***** (W) message." A complete list of error messages is included in Appendix G.

<u>The DELETE Function</u>

The DELETE function is used to delete records from the file. Data cards for this particular transaction type are prepared in the format:

```
 1-15 -- Key of record being deleted
16-80 -- Blank
```

The INSERT Function

The INSERT function is used to add records to the file. Data cards are prepared as directed in Appendix G.

The REWRITE Function

The REWRITE function is used to alter any fields within existing records other than the key field. Data cards are prepared as in Appendix G, leaving those fields which are not to be changed blank. Any combination of the three data cards may be used to rewrite a record. On each card included code the card code, the key, and any fields being altered. To replace a character field with blanks, code dollar signs throughout the field.

The NEW-KEY Function

The NEW-KEY function is used to alter a record's existing key field. Data cards are prepared in the format:

```
 1-15 -- Key of existing record
   16 -- Equal sign
17-31 -- New key
32-80 -- Blank
```

COPY

COPY is used to generate a backup copy of the file. Its use requires knowledge of OS/VS1 Job Control Language. A SAVERRX DD statement is supplied to allocate a tape or disk output file for use by COPY. The DCB parameters LRECL=185 and RECFM=FB are always coded. A blocksize that is a multiple of 185 must also be supplied. An example run is:

```
// JOB
// EXEC HISRRX,DISP=OLD
//SAVERRX DD UNIT=TAPE,VOL=SER=888888,
//           DISP=(NEW,KEEP),DSNAME=HIS.RAILROAD.BACKUP,
//           DCB=(BLKSIZE=18500,LRECL=185,RECFM=FB)
//SYSIN DD *
:COPY,FILE=RAILROAD
/*
```

CREATE

CREATE reloads the file from a backup copy. An example run is:

```
// JOB
// EXEC HISRRX,DISP=OLD
//SAVERRX DD UNIT=TAPE,VOL=SER=888888,DISP=OLD,
//           DSNAME=HIS.RAILROAD.BACKUP
//SYSIN DD *
:CREATE,FILE=RAILROAD
/*
```

REORGANIZE

File updates result in wasted storage space, and periodic reorganization is necessary. Reorganization is performed by generating a backup copy and then reloading. This can be performed in two steps by using COPY and CREATE, or in one step using REORGANIZE. An example run is:

```
// JOB
// EXEC HISRRX,DISP=OLD
//SAVERRX DD UNIT=TAPE,VOL=SER=888888,
//           DISP=(NEW,KEEP),DSNAME=HIS.RAILROAD.BACKUP,
//           DCB=(BLKSIZE=18500,LRECL=185,RECFM=FB)
//SYSIN DD *
:REORGANIZE,FILE=RAILROAD
/*
```

The Railroad Report File

The railroad report file is used for producing reports about crossings. The file is generated from the railroad, roadlog, traffic, and true mileage files.

CREATE-RRXREP

CREATE-RRXREP creates the railroad report file. No parameters are required. An example run is:

```
// JOB
// EXEC HISRRX,DISP=OLD
//SYSIN DD *
:CREATE-RRXREP
/*
```

RRXREP-SORT-LIST

RRXREP-SORT-LIST sorts the report file according to one of four different hierarchies, and then lists the result. These hierarchies are:

SYSTEM / ROUTE / MILEPOST

SYSTEM / ROUTE / ASCENDING ADJUSTED HAZARD INDEX / MILEPOST

SYSTEM / DESENDING ADJUSTED HAZARD INDEX / ROUTE / MILEPOST

OPERATING RAILROAD / DESCENDING ADJUSTED HAZARD INDEX / ROUTE / MILEPOST

These different methods of sorting are referenced, respectively, by the required parameter:

$$\text{LIST=} \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \right\}$$

The following parameters may optionally be coded:

```
        ⎧ ⎧ INTERSTATE                     ⎫ ⎫
        ⎪ ⎪ PRIMARY                        ⎪ ⎪
        ⎪ ⎨ SECONDARY      [=n [-n]]       ⎬ ⎪
        ⎪ ⎪ URBAN                          ⎪ ⎪
DATA=   ⎨ ⎩ LOCAL                          ⎭ ⎬
        ⎪ ⎧ INTERSTATE+PRIMARY ⎫             ⎪
        ⎪ ⎨ FEDERAL-AID        ⎬             ⎪
        ⎩ ⎩ ALL                ⎭             ⎭
```

```
START-MILEPOINT=xxx+x.xxx

END-MILEPOINT=xxx+x.xxx
```

An example run is:

```
// JOB
// EXEC HISRRX
//SYSIN DD *
:RRXREP-SORT-LIST,LIST=1
:RRXREP-SORT-LIST,LIST=2,DATA=PRIMARY=8-10
/*
```

APPENDIX A

CODING AND EDITING OF

ROADLOG DATA CARDS

When inserting or rewriting roadlog records, data cards are prepared as directed in this appendix.

Three data card types are used in the roadlog subsystem. The card type is identified by an "A," a "B," or a "C" in column 1.

When inserting a descriptor record, only an A card is used, and the only fields coded are the card type in column 1, the key, the description, and one of the remark codes DS, EN, ER, CO or IL.

When inserting a mileage record, an A card and a B card are always required. A C card is optional.

When rewriting a record, any combination of A, B, and C cards may be used. The card type and key must be coded. Remaining fields are coded only if being altered.

The A card format is:

Column(s)	Item
1	Card type -- "A"
2-16	Key
17-51	Description
52-55	Maintenance section
56-66	Project number
67-68	Remark
69-70	County number
71-72	Forest highway number
73-74	Administration code
75-76	Location code 1
77-78	Location code 2
79	Functional classification
80	Control of access

The B card format is:

Column(s)	Item
1	Card type -- "B"
2-16	Key
17	Number of lanes
18	Divided/undivided code
19	Population code
20-22	City number
23-24	Year built
25-26	Year improved
27-30	Surface type
31-32	Surface thickness
33-35	Base thickness
36-37	Surface width
38-39	Roadway width
40-44	Route length
45-49	Constructed length
50-54	Unimproved length
55-57	Wye length
58-62	Section length
63-68	Effective date
69	One-way code
70-71	Construction division
72-73	Planning division
74-75	Maintenance division
76-80	Defense section number

The C card format is:

Column(s)	Item
1	Card type -- "C"
2-16	Key
17-18	Section
19-21	Township
22-24	Range
25-26	Jurisdiction
27-28	Area name
29-30	Map sheet
31-80	Unused

The key consists of three subfields: a route system, a route number, and a milepoint. The route system is a 1-character system designation:

I -- Federal Aid Interstate
P -- Federal Aid Primary
S -- Federal Aid Secondary
U -- Federal Aid Urban
L -- Local

The route number is a right-justified 5-digit value; leading zeroes must be coded. The milepoint is nine characters long, consisting of a 3-digit reference post number and a 6-character distance from the reference post. The reference post is right-justified, and leading zeroes must be coded. The distance is coded in the format "+n.nnn," and is the distance in miles from the specified reference post to the beginning of the roadlog section.

The description field contains a verbal description. When coded on CO or IL records, the description takes on a fixed format indicating the location of the coincident section or the interstate loop. On a CO record, the first four characters of the description are the letters "COIN." On an IL record, the first four characters are "LOOP." The description is coded as:

1-4	COIN or LOOP
5	Blank
6	Route system (I, P, S, U, or L)
7-11	Route number
12	Blank
13-21	Beginning milepoint
22	Hyphen
23-31	Ending milepoint
32-35	Blank

The maintenance section is coded as a four-digit right-justified number.

The project number and administration code fields are closely related. If not applicable for a section, both fields are left blank. Otherwise, the project number field is coded as a 1-,2-,3- or 4-letter code, followed by at least one blank and descriptive information as required. The administration code is a 2-digit value that must correspond to the project number as defined within the project classification table (see programs LIST-PROJECT-TABLE and UPDATE-PROJECT-TABLE, documented in Chapter 2).

The remark code identifies the record type. For mileage records, the codes are:

LP -- Roadway section within a loop
SP -- Roadway section within a spur
NE -- Non-existent roadway section
OS -- Out-of-state roadway section
Blank -- Any other roadway section

For descriptor records, the remark codes are:

DS -- Signed route number and spur/loop descriptions
ER -- Additional descriptions
EN -- End-of-route
CO -- Coincident section
IL -- Interstate loop

The county number is coded as a two-digit number using the alphabetical numbering system:

01 -- Beaverhead	20 -- Granite	38 -- Powder River
02 -- Big Horn	21 -- Hill	39 -- Powell
03 -- Blaine	22 -- Jefferson	40 -- Prairie
04 -- Broadwater	23 -- Judith Basin	41 -- Ravalli
05 -- Carbon	24 -- Lake	42 -- Richland
06 -- Carter	25 -- Lewis and Clark	43 -- Roosevelt
07 -- Cascade	26 -- Liberty	44 -- Rosebud
08 -- Chouteau	27 -- Lincoln	45 -- Sanders
09 -- Custer	28 -- McCone	46 -- Sheridan
10 -- Daniels	29 -- Madison	47 -- Silver Bow
11 -- Dawson	30 -- Meagher	48 -- Stillwater
12 -- Deer Lodge	31 -- Mineral	49 -- Sweet Grass
13 -- Fallon	32 -- Missoula	50 -- Teton
14 -- Fergus	33 -- Musselshell	51 -- Toole
15 -- Flathead	34 -- Park	52 -- Treasure
16 -- Gallatin	35 -- Petroleum	53 -- Valley
17 -- Garfield	36 -- Phillips	54 -- Wheatland
18 -- Glacier	37 -- Pondera	55 -- Wibaux
19 -- Golden Valley		56 -- Yellowstone

The forest highway number is coded as a 2-digit right-justified number on mileage records corresponding to a forest highway, and is left blank on other mileage records.

Two location code fields are provided because many sections are applicable under two values. If only one code is applicable, the first field is coded and the second left blank. If two are applicable, the smaller value is coded in the first field and the larger in the second. The code values in use are:

01 -- City	09 -- Military reservation
02 -- Urban	10 -- State forest
03 -- County	11 -- State park
04 -- Indian reservation	12 -- State game preserve
05 -- National forest	13 -- Bureau of Land Management
06 -- National monument	
07 -- National park	
08 -- National wildlife refuge	

The functional classification is a 1-digit value:

1 -- Interstate	4 -- Major collector
2 -- Principal arterial	5 -- Minor collector
3 -- Minor arterial	6 -- Local roads

The control of access codes are:

1 -- No control
2 -- Partial control
3 -- Full control

The number of lanes is a one-digit numeric value.

The divided/undivided code is coded as "D" for divided highways and as blank for undivided highways.

The population code is coded on sections whose location is city or urban, and is a 1-digit value:

1 --	0-999
2 --	1,000-2,499
3 --	2,500-4,999
4 --	5,000-9,999
5 --	10,000-24,999
6 --	25,000-49,999
7 --	50,000 and over

The city number is coded on sections whose location is city or urban. The number is edited on input against the population code to ensure a match with the population specified for that city in the city name table (see programs LIST-CITY-TABLE and UPDATE-CITY-TABLE in Chapter 2). The city numbers are:

001 -- Alberton
002 -- Anaconda
003 -- Bainville
004 -- Baker
005 -- Bearcreek
006 -- Belgrade
007 -- Belt
008 -- Big Sandy
009 -- Big Timber
010 -- Billings
011 -- Boulder
012 -- Bozeman
013 -- Bridger
014 -- Broadus
015 -- Broadview
016 -- Brockton
017 -- Browning
018 -- Butte
019 -- Cascade
020 -- Chester
021 -- Chinook
022 -- Choteau
023 -- Circle
024 -- Clyde Park
025 -- Columbia Falls
026 -- Columbus
027 -- Conrad
028 -- Culbertson
029 -- Cut Bank
030 -- Darby
031 -- Deer Lodge
032 -- Denton
033 -- Dillon
034 -- Dodson
035 -- Drummond
036 -- Dutton
037 -- East Helena
038 -- Ekalaka
039 -- Ennis
040 -- Eureka
041 -- Fairfield
042 -- Fairview
043 -- Flaxville
044 -- Forsyth
045 -- Fort Benton
046 -- Froid
047 -- Fromberg
048 -- Geraldine
049 -- Glasgow
050 -- Glendive
051 -- Grassrange
052 -- Great Falls
053 -- Hamilton
054 -- Hardin
055 -- Harlem
056 -- Harlowton
057 -- Havre
058 -- Helena
059 -- Hingham
060 -- Hobson
061 -- Hot Springs
062 -- Hysham
063 -- Ismay
064 -- Joliet
065 -- Jordan
066 -- Judith Gap
067 -- Kalispell
068 -- Kevin
069 -- Laurel
070 -- Lavina
071 -- Lewistown
072 -- Libby
073 -- Lima
074 -- Livingston
075 -- Lodge Grass
076 -- Malta
077 -- Manhattan
078 -- Medicine Lake
079 -- Melstone
080 -- Miles City
081 -- Missoula
082 -- Moore
083 -- Nashua
084 -- Neihart
085 -- Opheim
086 -- Outlook
087 -- Philipsburg
088 -- Plains
089 -- Plentywood
090 -- Plevna
091 -- Polson
092 -- Poplar
093 -- Red Lodge
094 -- Rexford
095 -- Richey
096 -- Ronan
097 -- Roundup
098 -- Ryegate
099 -- Saco
100 -- St. Ignatius
101 -- Scobey
102 -- Shelby
103 -- Sheridan
104 -- Sidney
105 -- Stanford
106 -- Stevensville
107 -- Sunburst
108 -- Superior
109 -- Terry
110 -- Thompson Falls
111 -- Three Forks
112 -- Townsend
113 -- Troy
114 -- Twin Bridges
115 -- Valier
116 -- Virginia City
117 -- Walkerville
118 -- Westby
119 -- West Yellowstone
120 -- Whitefish
121 -- Whitehall
122 -- White Sulphur Springs
123 -- Wibaux
124 -- Winifred
125 -- Winnett
126 -- Wolf Point

The year built and year improved are coded as twentieth-century years (e.g., code 46 for 1946).

The surface type is a 4-digit value. A table of valid numbers is kept within the surface table (see programs LIST-SURFACE-TABLE and UPDATE-SURFACE-TABLE in Chapter 2).

The surface and base thicknesses are coded to the tenth of an inch. For example, code 3.7 inch surface thickness as 37, and 12.5 base thickness as 125.

Route mileage is coded on all mileage records (code 86.957 miles as 86957).

Constructed mileage is coded only on mileage records terminating a project. Code 2.356 miles as 02356.

Unimproved and wye mileages are coded on applicable records.

Section mileage is coded on all mileage records. This value is usually equal to route mileage, but may be less than route mileage because centerline-to-shoulder lengths are included in route mileage.

The effective date is the date from which the record is applicable. For example, if a road is built or substantially improved, and its completion date is July 9, 1972, code 070972.

The one-way code is 1 for one-way and 2 for two-way.

The construction division, planning division, and defense section number are not currently being coded; however, fields are available for use in the future.

The maintenance division is coded as one of the following values:

11	41
12	42
21	51
22	52
31	53
32	

Jurisdiction codes in use are:

01 -- County
02 -- National forest
03 -- Indian reservation
04 -- National park
05 -- National monument
06 -- National wildlife refuge
07 -- Bureau of Land Management
08 -- State park
09 -- State forest
12 -- State game preserve

Area names in use are:

65 -- Beaverhead National Forest
66 -- Bitterroot National Forest
67 -- Coeur d'Alene National Forest
68 -- Custer National Forest
69 -- Deerlodge National Forest
70 -- Flathead National Forest
71 -- Gallatin National Forest
72 -- Helena National Forest
73 -- Kaniksu National Forest
74 -- Kootenai National Forest
75 -- Lewis and Clark National Forest
76 -- Lolo National Forest

78 -- Blackfoot Indian Reservation
79 -- Fort Belknap Indian Reservation
80 -- Northern Cheyenne Indian Reservation
81 -- Crow Indian Reservation
82 -- Flathead Indian Reservation
83 -- Fort Peck Indian Reservation
84 -- Rocky Boy Indian Reservation

85 -- Glacier National Park
86 -- Yellowstone National Park
87 -- National Bison Range

On input to UPDATE, all data cards are edited for possible errors as described in Chapter 2. A complete list of error messages that may be printed follows:

***** (E) CONVERSION ERROR -- NON-NUMERIC CHARACTER IN NUMERIC FIELD *****

A character other than blank or 0-9 appears within a numeric field.

***** (E) KEY IS IN ERROR *****

The key is not coded in the proper format.

***** (E) ERROR IN COINCIDENT DESCRIPTION FORMAT *****

The description coded in a CO or IL descriptor record does not conform to the proper format.

***** (E) ROUTE LENGTH IS ZERO *****

The route length is not coded on a mileage record.

***** (E) SECTION LENGTH IS ZERO *****

The section length is not coded on a mileage record.

***** (E) SECTION LENGTH EXCEEDS ROUTE LENGTH *****

The value coded within the section length field is larger than that in the route length field.

***** (E) PROJECT NUMBER IS MISSING *****

The project number is not coded on an in-state federal-aid mileage record.

***** (E) ADMINISTRATION CODE IS ZERO *****

The administration code is not coded on an in-state federal-aid mileage record.

***** (E) PROJECT NUMBER AND ADMINISTRATION CODE DO NOT AGREE *****

On a mileage record containing both a project number and an administration code, the two fields do not agree. The project classification table may require updating (see programs LIST-PROJECT-TABLE and UPDATE-PROJECT-TABLE in Chapter 2).

***** (E) DIVIDED/UNDIVIDED CODE IN ERROR *****

The divided/undivided code field contains an unknown code.

***** (E) NUMBER OF LANES IS MISSING *****

The number of lanes is not coded on a federal aid mileage record.

***** (E) ONE-WAY CODE IS IN ERROR *****

The one-way code is larger than 2.

***** (E) CITY NUMBER IS IN ERROR *****

The city number coded exceeds 126.

***** (E) CITY NUMBER AND POPULATION CODE DO NOT AGREE *****

This message is printed if the city number and population code do not agree. The city table may require updating (see programs LIST-CITY-TABLE and UPDATE-CITY-TABLE in Chapter 2). This message is printed if one of the two fields is coded and the other is left blank or coded as zero.

***** (E) COUNTY NUMBER IS IN ERROR *****

The county number exceeds 56, or is zero in an in-state mileage record.

***** (E) CITY NUMBER AND COUNTY NUMBER DO NOT AGREE *****

The city specified does not lie within the county specified.

***** (E) YEAR BUILT IS MISSING *****

The year built field is blank on a mileage record.

***** (E) YEAR BUILT EXCEEDS CURRENT YEAR *****

The year built field contains a value larger than the current year.

***** (E) YEAR IMPROVED IS MISSING *****

The year improved field is blank on a mileage record.

***** (E) YEAR IMPROVED EXCEEDS CURRENT YEAR *****

The year improved field contains a value larger than the current year.

***** (E) YEAR BUILT EXCEEDS YEAR IMPROVED *****

The year built value is larger than the year improved value.

***** (E) JURISDICTION IN ERROR *****

The jurisdiction field contains a value larger than 13.

***** (E) FIRST LOCATION CODE IN ERROR *****

The first location code field contains a value larger than 15 or a value of zero.

***** (E) SECOND LOCATION FIELD IN ERROR *****

The second location code field contains a value larger than 15.

***** (E) SURFACE WIDTH IS ZERO *****

The surface width is not coded on a federal aid mileage record.

***** (E) ROADWAY WIDTH IS ZERO *****

The roadway width is not coded on a federal aid mileage record.

***** (E) SURFACE WIDTH EXCEEDS ROADWAY WIDTH *****

The surface width field contains a value larger than the roadway width field.

***** (E) CONTROL OF ACCESS IN ERROR *****

The control of access field contains a value larger than 3.

***** (E) SURFACE TYPE IS ZERO OR IN ERROR *****

The surface type field is either zero on a federal aid mileage record or contains a value not found in the surface type table. It may be necessary to update the surface type table (see programs LIST-SURFACE-TABLE and UPDATE-SURFACE-TABLE in Chapter 2).

***** (E) FUNCTIONAL CLASSIFICATION IS IN ERROR *****

The functional classification field exceeds 6.

***** (E) MAINTENANCE DIVISION IS IN ERROR *****

The maintenance division contains a value other than is allowed.

***** (E) CONSTRUCTION DIVISION IS IN ERROR *****

The construction division field exceeds 12.

***** (E) PLANNING DIVISION IS IN ERROR *****

The planning division field exceeds 12.

***** (W) REMARK CODE IN ERROR *****

The remark field contains a code other than those allowed. The record is handled as a descriptor record.

***** (W) ONE OR MORE MILEAGE RECORD DATA ELEMENTS CODED ON DESCRIPTOR RECORD *****

One or more fields other than the card type, key, description and remark have been coded. The fields are retained within the record.

***** (W) CHECK NUMBER OF LANES: lanes *****

This message is printed whenever the number of lanes is coded as 1, 3, or larger than 4.

***** (W) SURFACE THICKNESS IS ZERO *****

This message is printed when the surface thickness is not coded on a federal aid mileage record.

***** (W) BASE THICKNESS IS ZERO *****

This message is printed when the base thickness is not coded on a federal aid mileage record.

***** (W) EFFECTIVE DATE IS NOT CODED *****

This message is printed whenever the effective date is not coded on a mileage record.

***** (W) FUNCTIONAL CLASSIFICATION IS ZERO *****

This message is printed when the functional classification is not coded on a federal aid mileage record.

APPENDIX B

CODING AND EDITING OF

TRUE MILEAGE DATA CARDS

When using the INSERT and REWRITE functions of the true mileage UPDATE program, data cards must be prepared as directed in this appendix. The true mileage data card format is:

Column(s)	Item
1-9	Key
10-15	True mileage
16-21	Effective date
22-80	Blank

The true mileage key consists of two subfields: a route number and a reference post. The route number consists of a 1-character system designation and a 5-digit route number within that system. The system codes in use are:

I -- Federal Aid Interstate
P -- Federal Aid Primary
S -- Federal Aid Secondary
U -- Federal Aid Urban
L -- Local

The reference post subfield is a 3-digit number.

The true mileage is coded as a 6-digit number, with a decimal point assumed (but not coded) between the third and fourth digits. For example, code a true mileage of 30.478 as 030478.

The effective date is coded as three 2-digit subfields: month, day, and year. For example, code June 8, 1974 as 060874.

On input to UPDATE, all data cards are edited for possible errors as described in Chapter 3. A complete list of error messages that may be printed follows.

***** (E) CONVERSION ERROR -- NON-NUMERIC CHARACTER IN NUMERIC FIELD *****

A non-numeric character appears within a numeric field.

***** (E) KEY IS IN ERROR *****

The key is not coded in the proper format.

***** (E) EFFECTIVE DATE IS IN ERROR *****

The value coded in the effective date field is not a valid date.

***** (W) EFFECTIVE DATE IS NOT CODED *****

The effective date field is blank or contains zeroes.

APPENDIX C

CODING AND EDITING OF TRAFFIC DATA CARDS

When inserting or rewriting traffic records, data cards must be prepared as directed in this appendix.

Two data card types are used in the traffic subsystem. The card type is identified by an "A" or a "B" in column 1. The A card is used for entering data for the three years preceding the current year. The B card is used for entering current year data. Either an A card, a B card, or both may be coded when inserting or rewriting a record. Several fields are duplicated on the two cards to allow use of either card alone. When both cards are supplied, these fields may be coded on either card. If coded on both cards, the fields coded on the B card are used.

The A card format is:

Column(s)	Item
1	Card type -- A
2-16	Key
17	A if actual count; E if estimated
18-33	Data for first year
34-49	Data for second year
50-65	Data for third year
66-68	Future factor
69-71	Design hour volume
72	Remark
73-78	Effective date
79-80	Blank

The B card format is:

Column(s)	Item
1	Card type -- B
2-16	Key
17	A if actual count; E if estimated
18-33	Data for current year
34-65	Blank
66-68	Future factor
69-71	Design hour volume
72	Remark
73-78	Effective date
79-80	Blank

The key consists of two subfields: a route number and a milepoint. The route number is six characters long, consisting of a 1-character system designation and a 5-digit route number within that system. The system codes accepted are:

I -- Federal Aid Interstate
P -- Federal Aid Primary
S -- Federal Aid Secondary
U -- Federal Aid Urban
L -- Local

The milepoint is nine characters long, consisting of a 3-digit reference post number and a 6-character distance from the reference post coded in the form "+n.nnn."

Data for each year is coded within a 16-character field. Each of these fields is broken down as follows:

1-2	Year
3-7	Average daily traffic
8-10	Percentage of out-of-state vehicles
11-13	Percentage of pickups
14-16	Percentage of commercial vehicles

The year is the twentieth century year (code 1974 as 74). The average daily traffic is a 5-digit number. The percentages are coded as 3-digit numbers (code 15.5% as 155).

The future factor and design hour volume are also 3-digit percentages, coded like the above percentages.

The remark field is coded with one of the following:

W -- Major rural break
T -- Major municipal break
N -- Major non-existent break
O -- Major out-of-state break
R -- Minor rural break
M -- Minor municipal break
C -- Coincident break
L -- Loop break
S -- Spur break

The effective date is coded as three 2-digit fields: month, day, and year. For example, code February 2, 1974 as 020274.

On input to UPDATE, all data cards are edited for possible errors as described in Chapter 4. A complete list of error messages follows.

***** (E) CONVERSION ERROR -- NON-NUMERIC CHARACTER IN NUMERIC FIELD *****

A character other than blank or 0-9 appears within a numeric field.

***** (E) KEY IS IN ERROR *****

The key is not coded in the proper format.

***** (E) ACTUAL/ESTIMATED CODE IS IN ERROR *****

The actual/estimated code field contains a character other than blank, A, or E.

***** (E) REMARK IS IN ERROR *****

The remark field contains a character other than those listed above.

***** (E) EFFECTIVE DATE IS IN ERROR *****

The value coded in the effective date field is not a valid date.

***** (W) EFFECTIVE DATE IS NOT CODED *****

The effective date is not coded on a traffic count record.

APPENDIX D

CODING AND EDITING OF

ACCIDENT DATA CARDS

When inserting or rewriting accidents, data cards must be prepared as directed in this appendix.

Five data card types are used in the accident subsystem. The card type is identified by a code in column 1:

Card type 1: "A" if investigated, "E" if uninvestigated.
Card type 2: "B" (only used for investigated accidents).
Card type 3: "C" if investigated, "G" if uninvestigated.
Card type 4: "D" if investigated, "H" if uninvestigated.
Card type 5: "I."

The five card types are referred to as A, B, C, D and I cards.

When inserting an accident, one A card must be coded. A B card is optional. One C-D card sequence must be coded for each vehicle and pedestrian involved in the accident. I cards are used to include additional injuries and fatalities than can be coded on the appropriate D card.

When rewriting an accident, all that must be coded is the record identification fields and any fields being changed. Instructions on coding the proper cards for rewriting are included in Chapter 5 under the program UPDATE.

The A card format is:

Column(s)	Item
1	Card type -- "A" or "E"
2	Sequence number -- code a zero
3-14	Accident number
15-24	Date and time of occurrence
25-27	City number
28-29	County number
30-41	Accident location
42-43	First hamful event

44-45	First object hit off roadway
46	Injury severity
47	Damage severity
48	Class of trafficway
49	Roadway-related location
50	Junction-related location
51-52	Number of vehicles
53-54	Number of pedestrians
55-56	Number of fatalities
57-58	Number of injuries
59	Weather condition
60	Road condition
61	Light condition
62-63	Traffic controls
64-65	Other damage -- type
66	Other damage -- severity
67	Other damage -- owner
68-69	Posted speed limit
70	Engineering study
71-72	Analysis -- field 1
73-74	Analysis -- field 2
75	Collision type
76-80	Blank

The B card format is:

Column(s)	Item
1	Card type -- "B"
2	Sequence number -- code a zero
3-14	Accident number
15-24	Date and time notified
25-34	Date and time arrived
35-80	Blank

The C card format is:

Column(s)	Item
1	Card type -- "C" or "G"
2	Sequence number
3-14	Accident number
15-36	Name
37-53	Driver license number
54-55	Driver license state
56-61	Date of birth
62	Re-examination code
63-68	Charge code
69-74	Summons number
75-80	Blank

The D card format is:

Column(s)	Item
1	Card type -- "D" or "H"
2	Sequence number
3-14	Accident number
15-19	Contributing circumstances
20-24	Driver
25-29	Passenger -- front center
30-34	Passenger -- front right
35-39	Passenger -- rear left
40-44	Passenger -- rear center
45-49	Passenger -- rear right
50-51	Vehicle number
52-53	Vehicle intent
54-55	Pedestrian number
56-57	Pedestrian intent
58-59	Body style
60	Trailer style
61-62	Vehicle year
63	Vehicle involved in interstate traffic
64-78	Vehicle license number or VIN
79	Damage level
80	Damage severity

The I card format is:

Column(s)	Item
1	Card type -- "I"
2	Sequence number
3-14	Accident number
15-19	Blank
20-24	Passenger
25-29	Passenger
30-34	Passenger
35-39	Passenger
40-44	Passenger
45-49	Passenger
50-63	Blank
64-78	Vehicle license or VIN
79-80	Blank

The sequence number is punched to allow sorting of cards in the event they are dropped; this field is not used within the Highway Information System.

The accident number is a unique 12-character identifier. For investigated accidents, the number is composed of the following subfields:

1-2	Year of accident
3-5	Investigating agency
6-8	Investigating officer
9-10	Month of accident
11-12	Accident number within month

The investigating agency is coded as zero for the Highway Patrol, and as the city number (see below) for city policemen. For uninvestigated accidents, the accident number is composed of the following subfields:

1-2	Year of accident
3	Rural/municipal code
4-5	Month of accident
6-8	County number (rural accidents) or city number (municipal accidents)
9-12	Sequence number within month

The rural/municipal code is "C" for rural and "M" for municipal. The city and county numbers are listed below.

The date and time are coded within a 10-character field:

1-2	Month
3-4	Day
5-6	Year
7-8	Hour
9-10	Minute

The time is coded on the 24-hour clock. Accidents occurring at midnight are coded as 2400. Accidents occurring between midnight and 1:00 A.M. are coded with an hour of zero (code 12:30 A.M. as 0030).

The city number is coded as zero for rural accients, and with the city number for municipal accidents. The city numbers are listed in Appendix A.

County numbers are coded on the vehicle registration numbering system rather than the alphabetical system used in other HIS subsystems. The numbers are:

01 -- Silver Bow	20 -- Valley	38 -- Glacier
02 -- Cascade	21 -- Toole	39 -- Fallon
03 -- Yellowstone	22 -- Big Horn	40 -- Sweet Grass
04 -- Missoula	23 -- Musselshell	41 -- McCone
05 -- Lewis and Clark	24 -- Blaine	42 -- Carter
06 -- Gallatin	25 -- Madison	43 -- Broadwater
07 -- Flathead	26 -- Pondera	44 -- Wheatland
08 -- Fergus	27 -- Richland	45 -- Prairie
09 -- Powder River	28 -- Powell	46 -- Granite
10 -- Carbon	29 -- Rosebud	47 -- Meagher
11 -- Phillips	30 -- Deer Lodge	48 -- Liberty
12 -- Hill	31 -- Teton	49 -- Park
13 -- Ravalli	32 -- Stillwater	50 -- Garfield
14 -- Custer	33 -- Treasure	51 -- Jefferson
15 -- Lake	34 -- Sheridan	52 -- Wibaux
16 -- Dawson	35 -- Sanders	53 -- Golden Valley
17 -- Roosevelt	36 -- Judith Basin	54 -- Mineral
18 -- Beaverhead	37 -- Daniels	55 -- Petroleum
19 -- Chouteau		56 -- Lincoln

The accident location field may take on four different formats. The first format is used for rural accidents occurring on the federal aid system:

1 -- Federal aid route system (I, P, S, or U)
2-4 -- Federal aid route number
5-7 -- Reference post
8-12 -- Distance from reference post in format "+nnnn."

For example, code a distance of 0.235 miles from reference post 30 on Primary route 8 as P008030+0235.

The second location field format is used for rural accidents occurring off the federal aid system:

1 -- "R"
2 -- Blank
3-4 -- Coordinates within section
5-7 -- Range
8-10 -- Township
11-12 -- Section

The third format is used for municipal accidents occurring on the federal aid system:

1 -- Federal aid route system (I, P, S or U)
2-4 -- Federal aid route number
5-8 -- X-coordinate
9-12 -- Y-coordinate

The final format is used for municipal off-system accidents:

1 -- "M"
2-4 -- City number or zeroes
5-8 -- X-coordinate
9-12 -- Y-coordinate

First harmful event codes are:

00 -- Not stated
01 -- Overturned
02 -- Other non-collision
03 -- Collision with pedestrian
04 -- Collision with motor vehicle in transport
05 -- Collision with motor vehicle in other roadway
06 -- Collision with parked motor vehicle
07 -- Collision with railway train
08 -- Collision with pedalcycle
09 -- Collision with animal
10 -- Collision with fixed object
11 -- Collision with other object

First object hit off roadway codes are:

00 -- No object hit (may also be coded as 24)
01 -- End of overpass or river crossing
02 -- Guardrail protecting overpass structure
03 -- Overpass railing or side of overpass

04 -- End of underpass
05 -- Pier of underpass
06 -- Guardrail protecting underpass
07 -- Lighting, power pole, signal pole
08 -- Guardrail protecting lighting or power pole
09 -- Sign
10 -- Guardrail protecting sign
11 -- Median guardrail
12 -- Guardrail along fill
13 -- End of guardrail
14 -- Other guardrail
15 -- Tree
16 -- Cut slope
17 -- Road approach
18 -- Rock or boulder
19 -- End of drainage pipe
20 -- Building or other structure
21 -- Fence
22 -- Raised median or curb
23 -- Other object
24 -- No object (may also be coded as 00)
25 -- Unknown

Injury severity codes are:

0 -- No injury
1 -- Fatal injury
2 -- Incapacitating injury (cannot normally perform)
3 -- Non-incapacitating injury (evidence of injury)
4 -- Possible injury (apparent symptoms)

Damage severity codes are:

0 -- No damage
1 -- Disabling damage
2 -- Functional damage
3 -- Other motor vehicle damage

Class of trafficway codes are:

1 -- Interstate
2 -- Other fully controlled access road
3 -- Other U.S. numbered route
4 -- Other state numbered route
5 -- Other major arterial
6 -- County road
7 -- Local street
8 -- Other road

Of these codes, 2 (other fully controlled access road) and 5 (other major arterial) are not currently being used in Montana.

Roadway-related location codes are:

1 -- On roadway
2 -- Off roadway

Junction-related codes are:

0 -- Non-junction
1 -- Intersection
2 -- Intersection-related
3 -- Driveway access

The number of vehicles field indicates the number of vehicles involved in the accident, and cannot be zero. One vehicle record must be coded for each vehicle indicated.

The number of pedestrians field indicates the number of pedestrians involved in the accident. One vehicle record must be coded for each pedestrian indicated.

The number of fatalities field indicates the total number of fatalities in the accident. The number of injuries field indicates the total number of injuries. These fields must equal the total number of injuries and fatalities indicated on D and I cards.

Weather condition codes are:

0 -- Not stated
1 -- Clear
2 -- Raining
3 -- Snowing
4 -- Fog
5 -- Other

Road condition codes are:

0 -- Not stated
1 -- Dry
2 -- Wet
3 -- Snowy
4 -- Icy
5 -- Other

Light condition codes are:

0 -- Not stated
1 -- Daylight
2 -- Dawn or dusk
3 -- Darkness, lighted
4 -- Darkness, unlighted
5 -- Other

Traffic control codes are:

00 -- No traffic control devices
01 -- Traffic signals
02 -- Traffic signals not working
03 -- Traffic signals with pedestrian heads
04 -- Traffic signals with pedestrian heads (heads not working)
05 -- Flasher
06 -- Flasher not working
07 -- Stop sign
08 -- Yield sign
09 -- Railraod signals
10 -- Railroad signals not working
11 -- Railroad gates
12 -- Railroad gates not working
13 -- Do not enter signs
14 -- Other regulatory sign
15 -- Warning sign
16 -- Pavement markings

Other damage type codes are:

00 -- No other damage
01 -- Signal, lighting, or power pole
02 -- Sign
03 -- Guardrail
04 -- Bridge
05 -- Building
06 -- Shrubbery or trees
07 -- Maintenance equipment
08 -- Fire hydrant
09 -- Road surface
10 -- Drainage structure
11 -- Fence
12 -- Barricades
13 -- Other

Other damage severity codes are:

0 -- No other damage
1 -- Minor damage
2 -- Moderate damage
3 -- Major damage

Other damage owner codes are:

0 -- No other damage
1 -- Federal
2 -- State
3 -- County
4 -- City
5 -- Private

When an engineering study is requested, code an "X" in the engineering study field. Otherwise, leave the field blank.

Analysis codes are:

00 -- No additional contributing circumstances
01 -- Failed to have vehicle under control (speed not involved)
02 -- Inattentive driving
03 -- Inexperience
04 -- Blackout, heart, stroke, etc.
05 -- Fell asleep
06 -- Sun glare
07 -- Raining
08 -- Snowing
09 -- Whiteout
10 -- Blowing snow
11 -- Whiteout -- meeting or following vehicle
12 -- Dust storm
13 -- Dust caused by wind or preceding vehicle on unoiled road surface
14 -- Road slippery or icy
15 -- Other weather conditions

16 -- Improper hitch
17 -- Blow out -- flat tire
18 -- Stone thrown by vehicle
19 -- Avoiding another vehicle
20 -- Avoiding pedestrian -- unexpected actions
21 -- Striking or avoiding domestic animal in roadway
22 -- Striking or avoiding wild animal in roadway
23 -- Striking or avoiding object in roadway
24 -- Distraction with vehicle
25 -- Distraction outside vehicle
26 -- Unwarranted slowing
27 -- Blinded by glaring lights other than vehicle
28 -- Passenger fell from vehicle
29 -- Occupant releases vehicle
30 -- Indian in violation on reservation -- Patrol has no jurisdiction
31 -- Traffic control sign -- missing, down, etc.
32 -- Wind blowing
33 -- Water on highway
34 -- Fog
35 -- Load shifted

Collision type codes are:

1 -- Head one
2 -- Rear end
3 -- Angle
4 -- Sideswipe meeting
5 -- Sideswipe passing
6 -- Backed into
7 -- Other

The date and time notified and date and time arrived fields are coded like the date and time occurred field above.

The name is coded last name first, than a comma, the first name, a blank, and the middle initial.

The driver license number is coded left-justified. The driver license state codes are:

AL -- Alabama
AK -- Alaska
AZ -- Arizona
AR -- Arkansas
CA -- California
CO -- Colorado
CT -- Connecticut
DE -- Delaware
DC -- District of Columbia
FL -- Florida
GA -- Georgia
GU -- Guam
HI -- Hawaii
ID -- Idaho
IL -- Illinois
IN -- Indiana
IA -- Iowa
KS -- Kansas
LA -- Louisiana
ME -- Maine
MD -- Maryland
MA -- Massachusetts
MI -- Michigan
MN -- Minnesota
MS -- Mississippi
MO -- Missouri
US -- U.S. Goverment
AB -- Alberta
BC -- British Columbia
MB -- Manitoba
NF -- Newfoundland
NK -- New Brunswick
MT -- Montana
NB -- Nebraska
NV -- Nevada
NH -- New Hampshire
NJ -- New Jersey
NM -- New Mexico
NY -- New York
NC -- North Carolina
ND -- North Dakota
OH -- Ohio
OK -- Oklahoma
OR -- Oregon
PA -- Pennsylvania
PR -- Puerto Rico
RI -- Rhode Island
SC -- South Carolina
SD -- South Dakota
TN -- Tennessee
TX -- Texas
UT -- Utah
VT -- Vermont
VI -- Virgin Islands
WA -- Washington
WV -- West Virginia
WI -- Wisconsin
WY -- Wyoming
NS -- Nova Scotia
OT -- Ontario
PE -- Prince Edward Island
QB -- Quebec
SK -- Saskatchewan

The re-examination code is an "X" if recommended for re-examination, and blank otherwise.

The contributing circumstances field consists of five subfields, in the following order:

Vision
Physical defects
Road defects
Mechanical defects
Possible violations

Vision codes are:

0 -- Vision not obscured	4 -- Smoke
1 -- Buildings	5 -- Dust
2 -- Trees or hedges	6 -- Other
3 -- Other vehicle	

Physical defects codes are:

0 -- No apparent defects	3 -- Illness
1 -- Vision	4 -- Missing limbs
2 -- Hearing	5 -- Other

Road defects codes are:

0 -- No road defects	3 -- Loose material
1 -- Holes or ruts	4 -- Construction
2 -- Shoulder	5 -- Other

Mechanical defects codes are:

0 -- No apparent defects	3 -- Tires or steering
1 -- Lights	4 -- Other
2 -- Brakes	

Possible violations codes are:

0 -- No apparent violations
1 -- Had been drinking
2 -- Reckless driving
3 -- Speed too fast for conditions
4 -- Fail to yield right-of-way

5 -- Improper passing
6 -- Improper backing
7 -- Improper turn
8 -- Fail to signal
9 -- Other

Driver and passenger fields consist of the following subfields:

1 -- Alcohol
2-3 -- Age
4 -- Sex
5 -- Injury severity

The alcohol field is 0 for had not been drinking and 1 for had been drinking. Code the age in the age field (code 00 for not stated, and 99 for age 100 or over). In the sex field, code M for male, F for female, and blank for unknown. In the injury severity field, code the same value as shown above for the injury severity field on the A card. When no one was in the driver or a passenger position, leave the five positions for that driver or passenger blank. When coding a pedestrian, code his information in the driver field, and leave the passenger fields blank.

The vehicle number field is the assigned vehicle number. Assign the number 1 to the first vehicle coded, the number 2 to the next, etc. Do the same for pedestrians.

Vehicle intent codes are:

00 -- Not stated (code 00 here when coding pedestrians)
01 -- Go straight ahead
02 -- Overtake
03 -- Make right turn
04 -- Make left turn

05 -- Make U turn
06 -- Slow or stop
07 -- Start in traffic lane
08 -- Start from parked position
09 -- Back
10 -- Remain stopped in traffic lane
11 -- Remain parked

Pedestrian intent codes are:

00 -- Not stated (code 00 when coding vehicles)
01 -- Crossing at intersection or in crosswalk
02 -- Crossing not at intersection or in crosswalk
03 -- Walking in roadway with traffic
04 -- Walking in roadway against traffic
05 -- Standing in roadway
06 -- Pushing or working on vehicle in roadway
07 -- Other working in roadway
08 -- Playing in roadway
09 -- Other in roadway
10 -- Not in roadway
11 -- Not stated (may be coded as 00)

Body style codes are:

00 -- Not stated
01 -- Passenger car
02 -- Mini bus or van
03 -- Bus
04 -- School bus
05 -- Pickup
06 -- Truck or truck tractor
07 -- Motor home
08 -- Motor cycle
09 -- Ambulance
10 -- Farm tractor or machinery
11 -- Construction machinery
12 -- Pickup with camper
13 -- Bicycle
14 -- Snowmobile
15 -- Other

Trailer style codes are:

0 -- No trailer
1 -- Camping trailer
2 -- Mobile home
3 -- Utility trailer
4 -- Boat trailer
5 -- Semi trailer
6 -- Commerical cargo trailer
7 -- Other trailer

The interstate traffic field is coded as "X" if the vehicle was involved in interstate traffic, and as blank otherwise.

If the vehicle license is available, code the license number in columns 64-71, the state (see above) in columns 72-73, and the license year in columns 74-75. Otherwise, code the vehicle identification number in columns 64-78.

Damage severity is coded as on A cards. Code an "X" in the damage level field if vehicle damage exceeds $250.

On input to UPDATE, all data cards are edited for possible errors as described in Chapter 5. Severe error messages that may be printed are:

"A" CARD MISSING

(Insert only) -- The first data card for an accident is not an A card.

SEQUENCE ERROR

A C card is not immediately followed by a D card (insert or rewrite), a D card is not immediately preceded by a C card (insert only), or a character other than A through I appears in column 1 (insert or rewrite).

YEAR IN ERROR

The date coded in the year occurred field does not match the year coded within the accident number.

COUNTY IN ERROR

A value of zero or a value of 57 or larger is coded.

NUMER OF VEHICLES IN ERROR

The number of vehicles coded does not match the number of C-D card sequences supplied for vehicles.

NUMBER OF PEDESTRIANS IN ERROR

The number of pedestrians coded on the A card does not match the number of C-D card sequences supplied for pedestrians.

KEYPUNCH ERROR

This message is printed for any of these errors:

1. A character other than blank or 0-9 in a numeric field.
2. Both a vehicle number and a pedestrian number coded on a D card.
3. Neither a vehicle number nor a pedestrian number coded on a D card.

Warning messages that may be printed are:

MONTH OUT OF RANGE: VALUE CHANGED FROM xx TO 01

The month coded in the month of occurrence field is not in the range 01 to 12.

DAY OUT OF RANGE: VALUE CHANGED FROM xx TO 01

The day coded in the day of occurrence field is out of range.

CITY TOO LARGE: VALUE CHANGED FROM xxx TO ZERO

The city number exceeds 126.

COUNTY AND CITY DISAGREE: COUNTY CHANGED FROM xx TO yy

The city specified does not lie in the county specified.

INVALID FEDERAL AID ROUTE NUMBER PRESENT: VALUE LEFT AT xxxx

A non-existent route number is coded.

CITY PRESENT IN RURAL ACCIDENT: VALUE CHANGED FROM xxx TO ZERO

Both a city number and a federal aid milepoint are coded.

MILEPOST PRESENT BUT ROUTE SYSTEM IS NEITHER I, P, S NOR U

A milepoint is coded but the route system designation is unknown.

FIRST HARMFUL EVENT OUT OF RANGE: VALUE CHANGED FROM xx TO ZERO

The first harmful event field is larger than 11.

FIRST OBJECT OUT OF RANGE: VALUE CHANGED FROM xx TO 25

The first object hit off roadway contains a value larger than 25.

CLASS OF TRAFFICWAY IN ERROR: VALUE CHANGED FROM x TO y

The class of trafficway conflicts with the location field. Possible causes are:

1. Location specifies interstate, class of trafficway not 1.
2. Location specifies secondary, class of trafficway not 3 or 4.
3. Location specifies primary, class of trafficway not 3 or 4.
4. Class of trafficway is 0, 2, 5, or 9.

CLASS OF TRAFFICWAY IS 1, BUT MILEPOST IS NOT INTERSTATE

A class of trafficway of 1 is specified, but the location field does not contain an interstate location.

ROADWAY-RELATED LOCATION OUT OF RANGE: VALUE CHANGED FROM x TO 1

A value other than 1 or 2 is coded.

JUNCTION-RELATED LOCATION OUT OF RANGE: VALUE CHANGED FROM x TO ZERO

A value larger than 3 is coded.

WEATHER CONDITION OUT OF RANGE: VALUE CHANGED FROM x TO ZERO

A value larger than 5 is coded.

ROAD CONDITION OUT OF RANGE: VALUE CHANGED FROM x TO ZERO

A value larger than 5 is coded.

LIGHT CONDITION OUT OF RANGE: VALUE CHANGED FROM x TO ZERO

A value larger than 5 is coded.

TRAFFIC CONTROLS OUT OF RANGE: VALUE CHANGED FROM xx TO ZERO

A value larger than 16 is coded.

OTHER DAMAGE TYPE OUT OF RANGE: VALUE CHANGED FROM xx TO 13

A value larger than 13 is coded.

OTHER DAMAGE SEVERITY OUT OF RANGE: VALUE CHANGED FROM x TO 1

A value larger than 3 is coded.

OTHER DAMAGE OWNER OUT OF RANGE: VALUE CHANGED FROM x TO 5

A value larger than 5 is coded.

FIRST ANALYSIS FIELD OUT OF RANGE: VALUE CHANGED FROM xx TO ZERO

A value larger than 35 is coded.

SECOND ANALYSIS FIELD OUT OF RANGE: VALUE CHANGED FROM xx TO ZERO

A value larger than 35 is coded.

COLLISION TYPE OUT OF RANGE: VALUE CHANGED FROM x TO 7

The collision type field contains a value of zero or larger than 7.

NUMBER OF FATALITIES IN ERROR: VALUE CHANGED FROM xx TO yy

The number of fatalities coded on the A card does not agree with the number of fatalities indicated on the D and I cards.

NUMBER OF INJURIES IN ERROR: VALUE CHANGED FROM xx TO yy

The number of injuries coded on the A card does not agree with the number of injuries specified on the D and I cards.

DAMAGE SEVERITY IN ERROR: VALUE CHANGED FROM x TO y

The damage severity coded on the A card does not agree with the most severe damage specification on the D cards.

INJURY SEVERITY IN ERROR: VALUE CHANGED FROM x TO y

The injury severity coded on the A card does not agree with the most severe code on the D and I cards.

APPENDIX E

CODING AND EDITING OF SUFFICIENCY DATA CARDS

When inserting or rewriting records with UPDATE, data cards are prepared in accordance with this appendix. When rewriting records, code the key and any fields being altered. When inserting records, code the key and all applicable fields.

Descriptor records have only two fields coded: the key and the description. The description is one of the words "CITY," "COINCIDENT," "SPUR," "LOOP," "OUT-OF-STATE," or "END OF ROUTE." Rating records have all fields coded.

The sufficiency data card format is:

Column(s)	Item
1-15	Key -- Route system, route number, reference post, and distance
16-33	Description
34	Rural/Urban
35-36	Design speed
37	Terrain
38-39	Average speed
40-41	Percent of sight distance less than design
42-43	Number of stopping sight distances less than design
44-45	Number of horizontal curves sharper than design degree of curvature
46	Number of narrow bridges
47-48	Foundation rating
49-50	Surface rating
51-52	Drainage rating
53-59	[illegible]
60-65	Effective date

The rural-urban code is present on the coding form, but is not used in the present system.

The design speed for the Interstate sytem and designated portions of other Primary highways is 70 mph. For the Primary and Secondary highways classified as arterials, the normal range of design speed as

influenced by terrain conditions is from 40 to 60 mph. These conditions are:

Terrain	Design Speed
Level	60
Rolling	50-60
Mountainous	40-50

Frequent variation is design speeds on a sequence of rating sections is undesirable. Judgment must determine whether terrain conditions require reduced speeds by reason of non-feasibility of construction to higher standards. The following are design controls in relation to design speeds:

Design Speed (mph)	Max. Curve (degree)	Sight Distance (ft.)	
		Stopping	Passing
50	7½	350	1700
60	5	475	2000
70	3	600	2300

Two special design speeds are coded for under construction and non-existent sections. A code of 01 indicates a section of highway that is under construction. A code of 00 indicates a non-existent section.

Terrain is stored as a value of one through three:

1 -- Plains
2 -- Rolling
3 -- Mountainous

The classification "plains" means that no appreciable protion of the rating section contains natural features that would require extensive grading to attain minimum design requirements. "Rolling" country is that in which desirable design could feasibly be obtained even though a certain amount of heavy construction would be required. A "mountainous" classification indicates terrain of such character that design must be restricted considerably because of the non-feasibility of the construction effort required to raise the design speed.

The average highway speed is the weighted average of the design speeds within a highway section, when each subsection within the section is considered to have an individual design speed.

Sight distance is the distance visible to the driver of a passenger vehicle, measured along the normal travel path of a roadway, to the roadway surface or to a specified height above the roadway, when the view is unobstructed by traffic. For sections of two-lane primary routes and secondary arterials, it is necessary to ascertain the percent of the total rating-section length in which the road surface is not visible to the driver of a passenger car for a distance of at least 1500 feet, because of either horizontal or vertical curvature.

Stopping sight distance is the distance required by a driver of a vehicle, traveling at the design speed, to bring his vehicle to a stop after an object on the roadway becomes visible. It includes the distance traveled during the perception and reaction times and the vehicle braking distance.

The design degree of curvature is the greatest central angle subtended by a 100 foot length of curve which can be safely negotiated by a vehicle traveling at the highway design speed. The number of

horizontal curves sharper than the design degree of curvature indicates the number of curves within the sufficiency section which have a degree of curvature greater than the design degree of curvature.

A narrow bridge is any bridge within the sufficiency section which is narrower than the traveled-way width.

A foundation can only be rated 10 for being adequate or 0 for being inadequate. A rating of 0 is given to primary sections if any one of the following conditions are present:

1. Traveled-way less than 18 feet wide,
2. Lack of adequate and uniform cross section, including side ditches, or
3. Paved surface indicating failure which could not be corrected by the addition of a few inches of surfacing material.

For secondary sections, a rating of 0 is given if any one of the following conditions are present:

1. Traveled-way is less than 18 feet wide,
2. Not graded to reasonably adequate and uniform cross section, including side ditches,
3. Unpaved road showing signs of becoming impassable in adverse weather, or
4. Paved surface indicating failure than could not be corrected by the addition of a few inches of surfacing material.

For local-service routes, a rating of 0 is given if any one of the following conditions are present in the section:

1. Traveled-way less than 14 feet wide,
2. Unpaved road showing signs of becoming impassable in adverse weather, or
3. Paved surface indicating failure that could not be corrected by the addition of a few inches of surfacing material.

The highway surface is given a rating from 0 to 30. Certain controls pertaining to the remaining life of the pavement have been established to ensure a consistent surface rating system. When a paved surface is in relatively good condition, but showing first signs of failure (cracks, raveling, etc.), it should be rated 15. A surface of more advanced failure, while the road is still in fair and usable condition, will be rated between 10 and 15. A rating of 10 indicates that the pavement is in a condition that makes replacement incipiently justifiable. Increasingly poor conditions to the point of complete deterioration, creating a hazard, are rated 10 to 0. Rating above 15 is graduated in relation to surface smoothness, surface conformance to proper crown and grade, and uniformity of these conditions throughout the rating section. It should be kept in mind that the rating applies to the entire rating section. One short pavement failure (which can be remedied by a minor maintenance operation) should not significantly affect the section rating, therefore the rating is proportioned on the basis of the section length and failed length. Any considerable extent of variation in condition justifies an additional rating section. Gravel roads are rated only from 0 to 10 according to thickness, quality, and condition of the surfacing material. Unimproved, bladed, and graded and drained surface types are rated 0.

The sufficiency sections are rated from 0 to 10 depending upon the adequacy of drainage facilities. Lack or inadequacy of drainage facilities reduces the total of 10 points allotted for complete drainage.

The amount of reduction depends on the proportion of the total section length affected and the seriousness of the condition.

A section length must be coded on non-existent and under construction sections.

On input to UPDATE, all data cards are edited for possible errors as described in Chapter 6. Messages that may be printed are:

***** (E) CONVERSION ERROR -- NON-NUMERIC CHARACTER IN NUMERIC FIELD *****

A non-numeric character appears in at least one numeric field. The data card in error cannot be processed further, and no other edits are performed.

***** (E) KEY IS IN ERROR *****

The key is not coded in the proper format.

***** (E) DESIGN SPEED IS IN ERROR *****

The design speed does not have one of the values 0, 1, or 40-70.

***** (E) TERRAIN IS IN ERROR *****

The terrain is coded as a value other than 1, 2, or 3.

***** (E) AVERAGE SPEED IS IN ERROR *****

The average speed does not agree with the design speed.

***** (E) FOUNDATION RATING IS IN ERROR *****

The foundation rating coded exceeds 10.

***** (E) SURFACE RATING IS IN ERROR *****

The surface rating exceeds 30.

***** (E) DRAINAGE RATING IS IN ERROR *****

The drainage rating exceeds 10.

***** (E) SECTION LENGTH IS IN ERROR *****

Either a section length is coded on a record whose design speed is 40-70, or a section length is not coded on a rating whose design speed is 0 or 1 (non-existent and under construction sections).

***** (E) EFFECTIVE DATE IS IN ERROR *****

An invalid date is coded in the effective date.

***** (W) EFFECTIVE DATE IS MISSING *****

No effective date is coded.

APPENDIX F

CODING AND EDITING BRIDGE DATA CARDS

When inserting or rewriting bridge records, data cards are prepared as directed in this appendix.

Five data card types are used to code all of the bridge data fields. The first three card types are bridge description data. The remaining two are bridge appraisal data. The card type is indicated by a code of 1, 2, 3, 4, or 5 in column 17.

The card 1 format is:

Column(s)	Item Number	Item
1-16	1-1	Key
17	1-2	Card type -- "1"
18	1-3	Remark
19-34	1-4	Coincident bridge
35-36	1-5	Maintenance district
37-38	1-6	Construction district
39-41	1-7	Detour length
42-66	1-8	Features intersected
67-70	1-9	Minimum vertical clearance
71-73	1-10	Total horizontal clearance
74	1-11	Major or minor
75-80		Blank

The card 2 format is:

Column(s)	Item Number	Item
1-16	1-1	Key
17	1-2	Card type -- "2"
18-22	2-1	Latitude
23-28	2-2	Longitude
29-37	2-3	Inventory route
38-52	2-4	Facility carried
53	2-5	Physical vulnerability
54	2-6	Custodian
55-56	2-7	Year built
57-58	2-8	Year improved
59	2-9	Number of lanes on structure
60	2-10	Number of lanes under structure
61-64	2-11	Design load

65	2-12	Bridge median
66-67	2-13	Skew
68	2-14	Structure flared
69	2-15	Navigation control
70-72	2-16	Navigation vertical clearance
73-76	2-17	Navigation horizontal clearance
77-78	2-18	Type service
79-80		Blank

The card 3 format is:

Column(s)	Item Number	Item
1-16	1-1	Key
17	1-2	Card type -- "3"
18-20	3-1	Main structure type
21-23	3-2	Approach structure type
24-26	3-3	Number of spans in main unit
27-30	3-4	Number of approach spans
31-34	3-5	Length of maximum span
35-40	3-6	Structure length
41-43	3-7	Left sidewalk width
44-46	3-8	Right sidewalk width
47-50	3-9	Bridge roadway width
51-54	3-10	Bridge deck width
55-60	3-11	Station number
61-75	3-12	Project number
76-80		Blank

The card 4 format is:

Column(s)	Item Number	Item
1-16	1-1	Key
17	1-2	Card type -- "4"
18-21	4-1	Minimum vertical clearance over bridge roadway
22-25	4-2	Minimum vertical underclearance
26-28	4-3	Minimum lateral underclearance on right
29-31	4-4	Minimum lateral underclearance on left
32-35	4-5	Wearing surface
36-37	4-6	Wearing surface depth
38-41	4-7	Bridge approach guardrail
42-45	4-8	Main bridge guardrail

46-47	4-9	Posted speed limit
48-53	4-10	Posted load limit
54	4-11	Deck condition
55	4-12	Superstructure
56	4-13	Substructure
57	4-14	Channel and channel protection
58	4-15	Culvert and retaining walls
59-60	4-16	Estimated remaining life
61-63	4-17	Operating rating
64	4-18	Approach roadway alignment
65-67	4-19	Inventory rating
68	4-20	Structural condition
69	4-21	Deck geometry
70	4-22	Underclearances, vertical and horizontal
71	4-23	Safe load capacity
72	4-24	Waterway adequacy
73	4-25	Approach roadway alignment
74-80		Blank

The card 5 format is:

Column(s)	Item Number	Item
1-16	1-1	Key
17	1-2	Card type -- "5"
18-19	5-1	Year of needed improvement
20	5-2	Type of service
21-23	5-3	Type of work
24-29	5-4	Length of improvement
30	5-5	Proposed design loading of improvement
31-34	5-6	Proposed roadway width
35-36	5-7	Proposed number of lanes
37-42	5-8	Design ADT
43-44	5-9	Year of estimated ADT
45-46	5-10	Year of proposed adjacent roadway improvements
47	5-11	Type of proposed adjacent roadway improvements
48-52	5-12	Cost of improvements
53-58	5-13	Inspection date
59-64	5-14	Structure batch serial number
65-70	5-15	Microfilm serial number
71-80		Blank

The following pages describe each data item in more detail. Further details are available in the FHWA publication Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges dated July 1972. Items described in the FHWA publication are shown here with the item number in that publication.

Item 1-1 -- Key

The key is coded in identical format on all five card types. Column 1 is the route system code:

I -- Interstate
P -- Primary
S -- Secondary
U -- Urban
L -- Local

The route number is coded right-justified with leading zeroes included in columns 2-6. The reference post is coded right justified with all leading zeroes in columns 7-9. The distance from the reference post is coded in the format "+n.nnn" in columns 10-15. Column 16 is a sequence number to allow coding of several bridges at a given milepoint. Code 1 for the first or only bridge, 2 for the second, etc.

Item 1-2 -- Card Type

The card type indicates whether the card is a type 1, 2, 3, 4, or 5 card.

Item 1-3 -- Remark

The remark field is coded to indicate special conditions on a bridge. The following codes are in use:

A -- Adjacent opening of preceding structure
C -- Coincident route using same structure
D -- Indicates a dam
F -- Indicates a river ford
L -- Opposite traffic lane of preceding structure
P -- Parallel or dual structure
R -- Structure serving section direction traffic only
S -- Structure serving opposing traffic only
T -- Temporary structure
U -- Underpass

Item 1-4 -- Coincident Bridge

When a bridge is fully inventoried under another key, code the key of the record under which it is stored. All information on card type 1 must still be supplied, but cards 2-5 need not be coded.

Item 1-5 -- Maintenance District

The maintenance district is a two-digit code. Allowable values are:

11	41
12	42
21	51
22	52
31	53
32	

Item 1-6 -- Construction District

The construction district is a two-digit code.

Item 1-7 -- Detour Length -- FHWA Item 19

The detour length is a 3-digit field. If a ground level bypass is available, code zeroes. If the bridge is one of twin bridges and is not at an interchange, code 001 to indicate that the twin bridge can be used as a bypass. If neither of these is applicable, code the actual detour length to the nearest mile.

Item 1-8 -- Features Intersected -- FHWA Item 6

This field is a 25-character verbal description of features over or under the structure.

Item 1-9 -- Minimum Vertical Clearance -- FHWA Item 10

This is a four-digit field. The first two digits indicate feet, and the second two indicate inches. The value coded is the minimum clearance for the 10-foot width of traveled part of roadway where clearance is greatest. Code 9999 if no restriction exists.

Item 1-10 -- Total Horizontal Clearance -- FHWA Item 47

This 3-digit field gives the total horizontal clearance to the nearest tenth of a foot. Clearance is measured between the most restrictive features.

Item 1-11 -- Major or Minor

Code a Y for a major structure, and an N for a minor structure.

Item 2-1 -- Latitude -- FHWA Item 16

The latitude is a five digit field, coded in degrees, minutes and tenths of minutes. For example, code 35° 27.3' as 35273.

Item 2-2 -- Longitude -- FHWA Item 17

The longitude is a 6-digit field, and is identical to the latitude field except that three digits are required for degrees.

Item 2-3 -- Inventory Route -- FHWA Item 5

The inventory route is a 9-digit field describing the route carried by the structure or the route going under the structure. The first digit indicates whether the route is carried by the structure or goes under it:

1 -- Route carried by structure
2 -- Route under structure

The second digit indicates the kind of roadway:

1 -- Interstate
2 -- U.S. numbered highway
3 -- State numbered highway
4 -- County highway
5 -- City street
6 -- Federal lands road
7 -- State lands road
8 -- Other

The third digit indicates special conditions:

0 -- No special conditions
1 -- Mainline
2 -- Alternate
3 -- Bypass
4 -- Spur
5 -- Toll roads
6 -- Business
7 -- Ramp or wye
8 -- Service and/or unclassified frontage road
9 -- Truck route

The route number is coded right-justified in digits 4-8. The last digit indicates direction:

0 -- Not applicable
1 -- North
2 -- East
3 -- South
4 -- West

Item 2-4 -- Facility Carried

This field is a 15-character verbal description of the facility carried by the structure.

Item 2-5 -- Physical Vulnerability -- FHWA Item 18

This field is a 1-digit code:

0 -- No structure
1 -- Timber trestle
2 -- Concrete girder
3 -- Steel girder
4 -- Cantilever and truss
5 -- Suspension
6 -- Reinforced concrete -- massive arch
7 -- Dam bridge
8 -- Box culverts
9 -- Tunnels

Item 2-6 -- Custodian -- FHWA Item 21

This field is a 1-digit code indicating the type of agency responsible for maintaining the structure:

1 -- State highway department
2 -- Other state agency
3 -- County agency
4 -- City or other local agency
5 -- Federal agency
6 -- Railroad
7 -- Other private
8 -- Combination
9 -- Unknown

Item 2-7 -- Year Built -- FHWA Item 27

Code the year the structure was built.

Item 2-8 -- Year Improved -- FHWA Item 27

Code the latest year of major reconstruction.

Item 2-9 -- Number of Lanes on Structure -- FHWA Item 28

One-digit number of lanes.

Item 2-10 -- Number of Lanes Under Structure -- FHWA Item 28

One-digit number of lanes.

Item 2-11 -- Design Load -- FHWA Item 31

This four-digit field indicates the HS rating of the bridge. Code the H value in the first two digits, and the S value in the last two digits. Several special codes are used:

0000 -- Unknown bridge type
0007 -- Pedestrian bridge
0008 -- Railroad bridge
0009 -- Other bridge type

Item 2-12 -- Bridge Median -- FHWA Item 33

Code a 1-digit value:

0 -- No median
1 -- Open median
2 -- Closed median

Item 2-13 -- Skew -- FHWA Item 34

Code the skew angle to the nearest degree. Code a value of 99 if there is a major variation in skews of sub-structure units.

Item 2-14 -- Structure Flared -- FHWA Item 35

Code a 1 if the width of the structure varies, and a 0 if the width does not vary.

Item 2-15 -- Navigation Control -- FHWA Item 38

Code a 1 if navigation control exists, and a zero if it doesn't.

Item 2-16 -- Navigation Vertical Clearance -- FHWA Item 39

Code the vertical clearance to nearest foot if navigation control exists.

Item 2-17 -- Navigation Horizontal Clearance -- FHWA Item 40

Code the horizontal clearance to nearest foot if navigation control exists.

Item 2-18 -- Type Service -- FHWA Item 42

This 2-digit field indicates both the type of service on the bridge and the type of service under the bridge. The first digit indicates service on the bridge:

1 -- Highway
2 -- Railroad
3 -- Pedestrian exclusively
4 -- Highway and railroad
5 -- Highway and pedestrian
6 -- Overpass structure at an interchange or second level of multilevel interchange
7 -- Third level of multilevel interchange
8 -- Fourth level of multilevel interchange
9 -- Building or plaza
0 -- Other

The second digit indicates type of service under the bridge:

1 -- Highway, with or without pedestrian
2 -- Railroad
3 -- Pedestrian exclusively
4 -- Highway and railroad
5 -- Waterway
6 -- Highway and waterway
7 -- Railroad and waterway
8 -- Highway, railroad, and waterway
9 -- Relief
0 -- Other

Item 3-1 -- Main Structure Type -- FHWA Item 43

The first digit of this 3-digit field indicates the type of design and kind of material:

1 -- Concrete
2 -- Concrete continuous
3 -- Steel
4 -- Steel continuous
5 -- Prestress concrete
6 -- Prestress concrete continuous
7 -- Timber
8 -- Masonry
9 -- Aluminum, W.I., or C.I.
0 -- Other

The second and third digits indicate the type of design and/or construction:

01 -- Slab
02 -- Stringer/multi-beam or girder
03 -- Girder and floorbeam system
04 -- Tee beam
05 -- Box beam or girders -- multiple
06 -- Box beam or girders -- single or spread
07 -- Frame
08 -- Orthotropic
09 -- Truss -- Deck
10 -- Truss -- thru
11 -- Arch -- deck
12 -- Arch -- thru
13 -- Suspension
14 -- Stayed girder
15 -- Movable -- lift
16 -- Movable -- bascule
17 -- Movable -- swing
18 -- Tunnel
19 -- Culvert
00 -- Other

Item 3-2 -- Approach Structure Type -- FHWA Item 44

Codes for this item are as for the main structure type (items 3-1). Code zeroes throughout the field if the field is not applicable. If design and/or material is varied in the approach structure, code the first digit as zero. If no one type of design and/or construction is predominate in the approach units, code 20 in the last two digits.

Item 3-3 -- Number of Spans in Main Unit -- FHWA Item 45

Code the number of spans in the main or major unit.

Item 3-4 -- Number of Approach Spans -- FHWA Item 46

Code the number of spans in the approach spans to the major bridge.

Item 3-5 -- Length of Maximum Span -- FHWA Item 48

Code the maximum span length to the nearest foot.

Item 3-6 -- Structure Length -- FHWA Item 49

Code the structure length to the nearest foot.

Item 3-7 -- Left Sidewalk Width -- FHWA Item 50

Code the sidewalk width to the nearest tenth of a foot.

Item 3-8 -- Right Sidewalk Width -- FHWA Item 50

Code the sidewalk width to the nearest tenth of a foot.

Item 3-9 -- Bridge Roadway Width -- FHWA Item 51

Code the curb to curb width to the nearest tenth of a foot.

Item 3-10 -- Bridge Deck Width -- FHWA Item 52

Code the out-to-out deck width to the nearest tenth of a foot.

Item 3-11 -- Station Number

Code the station number to the nearest foot of the beginning of the bridge. Code a bridge beginning at station 740+00 as 074000.

Item 3-12 -- Project Number

Code in this 15-character field the project number under which the bridge was constructed.

Item 4-1 -- Minimum Vertical Clearance Over Bridge Roadway -- FHWA Item 53

Code the minimum vertical clearance to any superstructure restriction. Code feet in the first two digits, and inches in the last two digits. If no restriction exists, code 9999.

Item 4-2 -- Minimum Vertical Underclearance -- FHWA Item 54

Code the minimum clearance from the roadway or railroad track under the structure to the underside of the superstructure in feet and inches. Code zeroes if the structure is over any feature other than a roadway or railroad track.

Item 4-3 -- Minimum Lateral Underclearance on Right -- FHWA Item 55

If the structure is over a roadway or railroad, code the lateral underclearance to the nearest tenth of a foot. Otherwise, code 999.

Item 4-4 -- Minimum Lateral Underclearance on Left -- FHWA Item 56

This field is coded like item 4-3 if the structure is over a divided highway. Otherwise, code 999 in this field.

Item 4-5 -- Wearing Surface -- FHWA Item 57

This 4-digit field is coded with one of the following values:

0002 -- Open grate
0003 -- Wood planking
2010 -- Gravel or stone surface which consists of gravel, slag, broken stone, chert, caliche iron ore, chat, disintegrated rock or granite, or other similar fragmented material
4131 -- Asphalt
7001 -- Portlant cement concrete
9999 -- Other

Item 4-6 -- Wearing Surface Depth

Code the wearing surface thickness in inches.

Item 4-7 -- Bridge Approach Guardrail

The first two digits indicate the guardrail type. The second two digits indicate the guardrail height in inches. If there is no existing bridge approach guardrail leave both fields blank.

Item 4-8 -- Main Bridge Guardrail

The first two digits indicate the guardrail type. The remaining two digits give the guardrail height in inches.

Item 4-9 -- Posted Speed Limit

Code the speed limit posted for the bridge. If there is no speed limit posted, leave this field blank.

Item 4-10 -- Posted Load Limit

Code the posted load limit in tons. If there is no posted load limit, leave this field blank.

Item 4-11 -- Deck Condition -- FHWA Item 58

Code one of the following:

N -- Not applicable
9 -- New condition
8 -- Good condition -- no repair necessary
7 -- Minor items in need of repair by maintenance forces
6 -- Major items in need of repair by maintenance forces
5 -- Major repair contract needs to be let
4 -- Minimum adequate to tolerate present traffic, immediate rehabilitation necessary to keep open
3 -- Inadequacy to tolerate present heavy load -- warrants closing bridge to trucks
2 -- Inadequacy to tolerate any live load -- warrants closing bridge to all traffic
1 -- Bridge repairable, if desirable to reopen to traffic
0 -- Bridge conditions beyond repair -- danger of immediate collapse

Item 4-12 -- Superstructure -- FHWA Item 59

Codes are as for item 4-11.

Item 4-13 -- Substructure -- FHWA Item 60

Codes are as for item 4-11.

Item 4-14 -- Channel and Channel Protection -- FHWA Item 61

Codes are as for item 4-11.

Item 4-15 -- Culvert and Retaining Walls -- FHWA Item 62

Codes are as for item 4-11.

Item 4-16 -- Estimated Remaining Life -- FHWA Item 63

Code the estimated number of years of remaining life without major reconstruction.

Item 4-17 -- Operating Rating -- FHWA Item 64

The first digit indicates the type of loading as follows:

1 -- H truck
2 -- HS truck
3 -- Alternate interstate loading
4 -- 3-axle truck (type 3)
5 -- 3-S semi-trailer
6 -- 3-3 trailer
7 -- Railroad loading
8 -- Pedestrian or special loading
9 -- Gross load only given

The remaining two digits give the gross loading in tons, except for pedestrian and railroad loading. For railroad loading, code the Cooper Class or equivalent. Code pedestrian loading as 800.

Item 4-18 -- Approach Roadway Alignment -- FHWA Item 65

Codes are as for item 4-11.

Item 4-19 -- Inventory Rating -- FHWA Item 66

Codes are as for item 4-17.

Item 4-20 -- Structural Condition -- FHWA Item 67

Code one of the following:

N -- Not applicable
9 -- Conditions superior to present desirable criteria
8 -- Conditions equal to present desirable criteria
7 -- Condition better than present minimum criteria
6 -- Condition equal to present minimum criteria
5 -- Condition somewhat better than minimum adequacy to tolerate being left in place as is
4 -- Condition meeting minimum tolerable limits to be left in place as is
3 -- Basically intolerable condition requiring high priority of repair
2 -- Basically intolerable condition requiring high priority of replacement
1 -- Immediate repair necessary to put back in service
0 -- Immediate replacement necessary to put back in service

Item 4-21 -- Deck Geometry -- FHWA Item 68

Codes are as for item 4-20.

Item 4-22 -- Underclearances, Vertical and Horizontal -- FHWA Item 69

Codes are as for item 4-20.

Item 4-23 -- Safe Load Capacity -- FHWA Item 70

Codes are as for item 4-20.

Item 4-24 -- Waterway Adequacy -- FHWA Item 71

Codes are as for item 4-20.

Item 4-25 -- Approach Roadway Alignment -- FHWA Item 72

Codes are as for item 4-20.

Item 5-1 -- Year of Needed Improvement -- FHWA Item 73

Code the year in which improvements are estimated to be needed. Coded zeroes if no improvements are needed.

Item 5-2 -- Type of Service -- FHWA Item 74

Use the code in item 2-18 to represent type of service to be provided on the bridge.

Item 5-3 -- Type of Work -- FHWA Item 75

The following codes indicate type of work proposed:

30 -- Widening existing bridge or other major structure
31 -- Replacement of bridge or other structure because of condition
32 -- Replacement of bridge or other structure because of relocation of road
33 -- Construction of new bridge or major structure (except to eliminate a railroad grade crossing or one for pedestrians only)

34 -- Construction of pedestrian overcrossing or undercrossing
35 -- Other structure work
36 -- Strengthening
37 -- Rehabilitation

In addition, a one-digit suffix code indicates whether work is to be done by force account or by contract:

1 -- Contract
2 -- State forces

Item 5-4 -- Length of Improvement -- FHWA Item 76

Code improvement length to nearest foot.

Item 5-5 -- Proposed Design Loading of Improvement -- FHWA Item 77

Code the proposed design loading as a 1-digit value:

1 -- H 10
2 -- H 15
3 -- HS 15
4 -- H 20
5 -- HS 20
6 -- HS 20+
7 -- Pedestrian
8 -- Railroad
9 -- Other
0 -- Unknown

Item 5-6 -- Proposed Roadway Width -- FHWA Item 78

Code the proposed width to the nearest foot.

Item 5-7 -- Proposed Number of Lanes -- FHWA Item 79

Code the proposed number of lanes.

Item 5-8 -- Design ADT -- FHWA Item 80

Code the design ADT if applicable.

Item 5-9 -- Year of estimated ADT -- FHWA Item 81

Code the year when the design ADT is estimated to exist.

Item 5-10 -- Year of Proposed Adjacent Roadway Improvements -- FHWA Item 82

Code the year in which it is expected that improvements to the roadway approaches to the bridge will take place.

Item 5-11 -- Type of Proposed Adjacent Roadway Improvements -- FHWA Item 83

Code a 1-digit value as follows:

0 -- Not applicable
1 -- Resurface
2 -- Reconstruction
3 -- Widening
4 -- Shoulder improvements
5 -- Other

Item 5-12 -- Cost of Improvements -- FHWA Item 84

Code the cost of proposed improvements to thousands of dollars.

Item 5-13 -- Inspection Date

Code the date the bridge was inspected.

Item 5-14 -- Structure Batch Serial Number

The structure batch serial number is the reference number for the structural data stored for the Control Data Corporation Bridge Analysis and Rating System. See "Bridge Analysis and Rating System: Manual 2 Data Preparation Instructions" by Control Data Corporation.

Item 5-15 -- Microfilm Serial Number

This number is used for locating microfilmed plans of the bridge.

On input to UPDATE, all data cards are edited for possible errors as described in Chapter 6. A complete list of error messages that may be printed follows:

*** ERROR - A NON-NUMERIC CHARACTER WAS DETECTED IN A NUMERIC FIELD

A character other than a numeric digit or a blank was coded in a numeric field.

*** ERROR - INVALID SYSTEM CODED

A route system code other than I, P, S, U, L, F, or R is coded.

*** ERROR - INVALID ROUTE NUMBER

A non-existent route number was coded.

*** ERROR - INVALID REFERENCE POST CODED

An invalid reference post was coded.

*** ERROR - INVALID MILEPOINT

An invalid milepoint was coded.

*** ERROR - INVALID SEQUENCE NUMBER CODED

A key sequence number other than a numeric digit was coded.

*** ERROR - INVALID REMARK FIELD CODED

A remark field other than C, U, A, P, R, S, L, T or blank was coded.

*** ERROR - BRIDGE REMARK INDICATES THAT A COINCIDENT KEY SHOULD BE CODED

A remark of C, U, S, or R is coded, but no coincident key is coded.

*** ERROR - INVALID COINCIDENT KEY CODED

The coincident key coded is not in the proper format.

*** ERROR - INVALID MAINTENANCE DISTRICT CODED

The maintenance district value is incorrect.

*** ERROR - INVALID CONSTRUCTION DISTRICT CODED

The construction district value is incorrect.

*** ERROR - INVALID VERTICAL CLEARANCE CODED

The vertical clearance coded is invalid.

*** ERROR - INVALID LATITUDE CODED

The latitude value is invalid.

*** ERROR - INVALID LONGITUDE CODED

The longitude value is invald.

*** ERROR - INVALID INVENTORY ROUTE CODED

The inventory route field is invalid.

*** WARNING - MORE THAN FOUR LANES ON THE STRUCTURE?

A value larger than 4 is coded for number of lanes on structure.

*** WARNING - MORE THAN FOUR LANES UNDER STRUCTURE?

A value larger than 4 is coded for number of lanes under structure.

*** ERROR - INVALID MEDIAN CODED

A median code larger than 2 is coded.

*** ERROR - INVALID BRIDGE FLARE CODED

The bridge flare is coded larger than 1.

*** ERROR - INVALID NAVIGATION CONTROL CODED

The navigation control contains a value larger than 1.

*** ERROR - AN ATTEMPT WAS MADE TO CODE NAVIGATION CLEARANCE WITHOUT NAVIGATION CONTROL

Either the vertical or the horizontal navigation clearance was coded, but the navigation control indicated no navigation.

*** ERROR - INVALID MAIN STRUCTURE TYPE CODED
*** ERROR - INVALID APPROACH STRUCTURE CODED

A structure number larger than 919 was coded.

*** ERROR - INVALID MINIMUM VERTICAL CLEARANCE

The minimum vertical clearance is invalid.

*** ERROR - INVALID MINIMUM VERTICAL UNDERCLEARANCE

The minimum vertical underclearance is invalid.

*** ERROR - INVALID WEARING SURFACE CODED

The wearing surface is a value other than 2, 3, 2010, 4131, 7001, or 9999.

*** ERROR - INVALID APPROACH GUARDRAIL CODED

The first character of the approach guardrail is neither G nor 9.

*** ERROR - INVALID DECK CONDITION CODED
*** ERROR - INVALID SUPERSTRUCTURE CONDITION
*** ERROR - INVALID SUBSTRUCTURE CONDITION
*** ERROR - INVALID CHANNEL CONDITION CODED
*** ERROR - INVALID CULVERT CONDITION CODED
*** ERROR - INVALID APPROACH ALIGNMENT CONDITION
*** ERROR - INVALID STRUCTURE CONDITION CODED
*** ERROR - INVALID DECK GEOMETRY APPRAISAL
*** ERROR - INVALID APPRAISAL OF THE UNDERCLEARANCES
*** ERROR - INVALID APPRAISAL OF SAFE LOAD CAPACITY
*** ERROR - INVALID APPRAISAL OF THE WATERWAY
*** ERROR - INVALID APPRAISAL OF THE APPROACH ROADWAY ALIGNMENT

A value other than N or a decimal digit is coded.

*** ERROR - INVALID TYPE OF WORK CODED

The type of work was not zero, but was less than 130 or greater than 237.

*** WARNING - NUMBER OF PROPOSED LANES IS MORE THAN 4?

The number of proposed lanes was coded as larger than 4.

*** ERROR - INVALID TYPE OF IMPROVEMENT CODED

The type of improvement coded exceeded 5.

*** ERROR - FAILURE TO CODE INSPECTION DATE

The inspection date was not coded.

APPENDIX G

CODING AND EDITING RAILROAD DATA CARDS

When inserting or rewriting railroad file records, data cards are prepared as directed in this appendix. Three data cards are used to code all the railroad data fields.

The first two cards are concerned with information gathered by a field survey team. The third card contains information that is to be taken from the D.O.T. - A.R.R. NATIONAL RAILROAD-HIGHWAY CROSSING INVENTORY. The card type is indicated with a code of A, B, or C in the first column of the card.

The A card format is:

Column(s)	Item #	Item
1	a-1	Card Code -- "A"
2-16	a-2	Key
17-22	a-3	Date of Field Survey
23-24	a-4	Road Width
25-49		(ROAD APPROACH ONE)
25	a-5	Direction
26-31		(Right Quadrant)
26-28	a-6	Angle at 0
29-31	a-7	Angle at P
32-37		(Left Quadrant)
32-34	a-8	Angle at 0
35-37	a-9	Angle at P
38-40	a-10	Distance OP
41-44	a-11	Curvature of Road
45-47	a-12	Grade of Road
48	a-13	Local Interference
49	a-14	Vertical Sight Restriction
50-74		(ROAD APPROACH TWO)
50	a-15	Direction
51-56		(Right Quadrant)
51-53	a-16	Angle at 0
54-56	a-17	Angle at N
57-62		(Left Quadrant)
57-59	a-18	Angle at 0
60-62	a-19	Angle at N
63-65	a-20	Distance ON
66-69	a-21	Curvature of Road
70-72	a-22	Grade of Road
73	a-23	Local Interference
74	a-24	Vertical Sight Restriction
75-80		Blank

The B card format is:

Column(s)	Item #	Item
1	b-1	Card Code -- "B"
2-16	b-2	Key
17-52		(RAILROAD TRAFFIC)
17-28		(Daily)
17	b-3	Number of Daily Passenger Trains
18-19	b-4	Maximum speed of Daily Passenger Trains
20-21	b-5	Number of Daily Freight Trains
22-23	b-6	Maximum Speed of Daily Freight Trains
24-26	b-7	Number of Daily Switch Trains
27-28	b-8	Maximum Speed of Daily Switch Trains
29-40		(Weekly)
29	b-9	Number of Weekly Passenger Trains
30-31	b-10	Maximum Speed of Weekly Passenger Trains
32-33	b-11	Number of Weekly Freight Trains
34-35	b-12	Maximum Speed of Weekly Freight Trains
36-38	b-13	Number of Weekly Switch Trains
39-40	b-14	Maximum Speed of Weekly Switch Trains
41-52		(Seasonal)
41	b-15	Number of Seasonal Passenger Trains
42-43	b-16	Maximum Speed of Seasonal Passenger Trains
44-45	b-17	Number of Seasonal Freight Trains
46-47	b-18	Maximum Speed of Seasonal Freight Trains
48-50	b-19	Number of Seasonal Switch Trains
51-52	b-20	Maximum Speed of Seasonal Switch Trains
53	b-21	Urban/Rural Code
54-78	b-22	Verbal Description of Location
79-80		Blank

The C card format is:

Column(s)	Item #	Item
1	c-1	Card Code -- "C"
2-16	c-2	Key
17-22	c-3	Date of DOT-AAR Crossing Inventory
23-26	c-4	Operating Railroad Co.
27-33	c-5	Crossing ID #
34-48	c-6	Branch or Line Name
49-55	c-7	Branch or Line Milepost
56	c-8	Number of Main Tracks
57-58	c-9	Number of Other Tracks
59	c-10	If Tracks Used by Other RR Co., Type of Use Code

60-63	c-11	Name of Other RR Co.
64	c-12	Number of Crossbucks at Crossing
65	c-13	Number of Stop Signs at Crossing
66	c-14	Number of Wigwags at Crossing
67	c-15	Number of Flashing Lights at Crossing
68	c-16	Number of Gates at Crossing
69	c-17	Number of Traffic Lanes at Crossing
70-72	c-18	Nightly Train Traffic
73-76	c-19	Estimated ADT (for local system roads)

The following pages describe each data item in more detail. Where an item is taken from the DOT-AAR Crossing Inventory, that item is marked by an @(m,n) with m being the roman numeral representing the part of that form, and n being the decimal number within that part from which the item is to be taken.

Item a-1 (Card Code)

An "A" is coded to indicate which data items the card contains.

Item a-2 (Key)

The key is coded in identical format on all three card types. Column 2 is the route system code:

I -- Interstate
P -- Primary
S -- Secondary
U -- Urban
L -- Local

The route number is coded right-justified with leading zeros included in columns 3-7. The reference post number is coded right-justified with leading zeros included in columns 8-10. The distance from the reference post is coded in the format "+n.nnn" in columns 11-16.

Item a-3 (Date of Field Survey)

Code month, day, and year each right-justified in their respective two-character positions.

Item a-4 (Road Width)

Code the width of the road at the crossing in feet as a right-justified two-digit number.

Item a-5 (Direction)

This indicates the orientation of a driver as he crosses the crossing from road approach 1. Code in the following manner:

1 -- North
2 -- North-West
3 -- West
4 -- South-West
5 -- South
6 -- South-East
7 -- East
8 -- North-East

Item a-6 (Angle at 0)

This is the angle that is formed at the intersection of the tracks and the roadway in the right quadrant when travelling in the direction indicated by item a-5. Code as a right-justified three digit integer number of degrees.

Item a-7 (Angle at P)

This is the angle that is observed between the crossing and the first visual obstruction of the tracks in the right quadrant from a point 300' back from the crossing by road when travelling in the direction indicated by item a-5. Code as a right-justified three digit integer number of degrees.

Item a-8 (Angle at 0)

This is the angle that is formed at the intersection of the tracks and the roadway in the left quadrant when travelling in the direction indicated by item a-5. Code in same manner as item a-6.

Item a-9 (Angle at P)

This is the angle that is observed between the crossing and the first visual obstruction of the tracks in the left quadrant

from a point 300' back from the crossing by road when travelling in the direction indicated by item a-5. Code in same manner as item a-7.

Item a-10 (Distance OP)

This is the straight-line distance between the point 0, which coincides with the crossing, and the point P, which is located 300' back from the crossing by road when travelling in the direction indicated by item a-5. Code as a three digit right-justified integer number of feet.

Item a-11 (Curvature of Road)

This is the distance from the center of a 62' chord to the center of the roadway in tenths of feet, when both ends of the chord are in the center of the roadway. Code as a three digit integer number of tenths of feet preceeded by a plus sign "+" if the road curves to the right when travelling in the direction indicated by item a-5, or a minus sign "-" if the road curves to the left.

Item a-12 (Grade of Road)

This is the percentage grade of the road when travelling in the direction indicated by item a-5. Code as a two digit integer

percentage preceeded by a plus if the grade is upward, and a minus if the grade is downward.

Item a-13 (Local Interference)

If there is local interference within 150' of the crossing as approach is made in the direction indicated by item a-5, then code as follows:

0 -- none
1 -- busy driveway
2 -- intersecting street with no traffic signal
3 -- intersecting street with traffic signal

Item a-14 (Vertical Sight Restriction)

Code 0 (zero) if it is possible to see a 4.5' high vehicle on the other side of the crossing when 300' back from the crossing and making approach in the direction indicated by item a-5. Code 1 (one) if it is not possible.

Item a-15 (Direction)

This indicates the orientation of a driver as he crosses the crossing from road approach 2. Code as for a-5.

Items a-16 through a-24

These items are the same as items a-6 through a-14 and give information on the crossing when travelling as indicated by item a-15. Since the point which corresponds to P is, for the second direction, on the opposite side of the tracks, a different name is used for it: N. However, the meaning of the information is exactly the same. By coding a right and a left quadrant for both directions of approach, all four quadrants are covered.

Item b-1 (Card Code)

A "B" is coded to indicate which data items the card contains.

Item b-2 (key)

This is coded exactly as item a-2.

Item b-3 (Number of Daily Passenger Trains)

Code as a one digit integer, or leave blank.

Item b-4 (Maximum speed of Daily Passenger Trains)

Code as two digit right-justified integer.

Item b-5 (Number of Daily Freight Trains)

Code as two digit right-justified integer.

Item b-6 (Maximum Speed of Daily Freight Trains)

Code as two digit right-justified integer.

Item b-7 (Number of Daily Switch Trains)

Code as three digit right-justified integer.

Item b-8 (Maximum Speed of Daily Switch Trains)

Code as two digit right-justified integer.

Items b-9 through b-14 (Weekly Train Traffic)

All data on train traffic that occurs on a weekly basis should be coded using these items in the same manner as daily traffic was coded using items b-3 through b-8.

Items b-15 through b-20 (Seasonal Train Traffic)

All data on train traffic that occurs on a seasonal basis should be coded using these items in the same manner as daily traffic was coded using items b-3 through b-8.

Item b-21 (Urban/Rural Code)

Code "R" for Rural Crossing.

Code "U" for Urban Crossing.

Item b-22 (Verbal Description of Crossing Location)

Code left-justified this 25 character field.

Item c-1 (Card Code)

A "C" is coded to indicate which data items the card contains.

Item c-2 (Key)

This is coded exactly as item a-2.

Item c-3 (Date of Crossing Inventory)

Code in same manner as item a-3.

Item c-4 (Operating Railroad Co.)

Code left-justified the FRA code for the operating railroad. The name of the operating railroad can be found @(I,1).

Item c-5 (Crossing ID #)

Code left-justified the crossing ID # found @(I,11).

Item c-6 (Branch or Line Name)

Code left-justified the branch or line name found @(I,13).

Item c-7 (Branch or Line Milepost)

The format of this right-justified 7 digit field is "nnnn.nn". Can be found @(I,14).

Item c-8 (Number of Main Tracks)

Code as one digit integer found @(II,3).

Item c-9 (Number of Other Tracks)

Code as right-justified two digit integer found @(II,3).

Item c-10 (Type of Use Code)

This code is used to indicate the type of usage of the tracks by railroad companies. The information for coding can be found @(II,4-5). Code in the following manner:

0 (or blank) -- Only the Operating RR uses the tracks
1 -- another RR (other than operating RR) operates a separate track at the crossing
2 -- another RR (other than operating RR) operates a shared track at the crossing

Item c-11 (Name of Other Railroad)

Code left-justified the FRA code for the other railroad, or leave blank. The name of the other railroad can be found @(II,4-5).

Item c-12 (Number of Crossbucks)

Code as one digit integer, or leave blank. Found @(II,6).

Item c-13 (Number of Stop Signs)

Code in same manner as item c-12. Found @(II,6).

Item c-14 (Number of Wigwags)

Code in same manner as item c-12. Found @(II,6).

Item c-15 (Number of flashing Lights)

Code in same manner as item c-12. Found @(II,6).

Item c-16 (Number of Gates)

Code in same manner as item c-12. Found @(II,6).

Item c-17 (Number of Traffic Lanes)

Code as one digit integer. Found @(III,3).

Item c-18 (Nightly Train Traffic)

Code as a three digit right-justified integer. Found @(II,1).

Item c-19 (Estimated ADT)

This is for a crossing located in a section of roadway for which the traffic file has no ADT. Code as a 4 digit right-justified integer. Found @(IV,4).

On input to UPDATE, all cards are edited for possible errors. A complete list of error messages is as follows:

***** (E) INVALID KEY FIELD

A route system code other than I, P, S, U, or L was coded, or the key did not conform to the format given for item a-2.

***** (E) A NON-NUMERIC CHARACTER WAS DETECTED IN A NUMERIC FIELD

A character other than a numeric digit or a blank was coded in a numeric field.

***** (E) SURVEY DATE IN ERROR OR NOT CODED

The survey date was invalid, or not coded.

***** (E) INVENTORY DATE IN ERROR OR NOT CODED

The inventory date was invalid or not coded.

***** (E) ZERO ROAD WIDTH OR NOT CODED
***** (E) ZERO DIRECTION OR NOT CODED

Refers to item a-5 or a-15.

***** (E) ZERO ANGLE OR NOT CODED

Refers to items a-6, a-7, a-8, a-9, a-16, a-17, a-18, or a-19.

***** (E) ZERO DISTANCE OR NOT CODED

Refers to items a-10 or a-20.

***** (E) INVALID CODE FOR LOCAL INTERFERENCE
***** (E) INVALID CODE FOR VISUAL RESTRICTION
***** (E) INCONSISTANT RAILROAD TRAFFIC

Refers to an inconsistancy in the coding of railroad traffic, items b-3 through b-20.

***** (E) ZERO NUMBER OF MAIN TRACKS OR NOT CODED
***** (E) TYPE OF USE NOT CONSISTANT WITH NUMBER OF TRACKS

Refers to items c-8 through c-10.

***** (E) TYPE OF USE NOT CONSISTANT WITH OTHER RAILROAD NAME

Refers to items c-10 and c-11

***** (E) ZERO NUMBER OF TRAFFIC LANES OR NOT CODED
***** (E) NIGHTLY TRAINS GREATER THAN TOTAL DAILY TRAINS

Refers to items b-3 and c-18.

***** (E) URBAN/RURAL NOT CODED
***** (E) ESTIMATED ADT NOT CODED FOR A LOCAL CROSSING

Refers to item c-19.